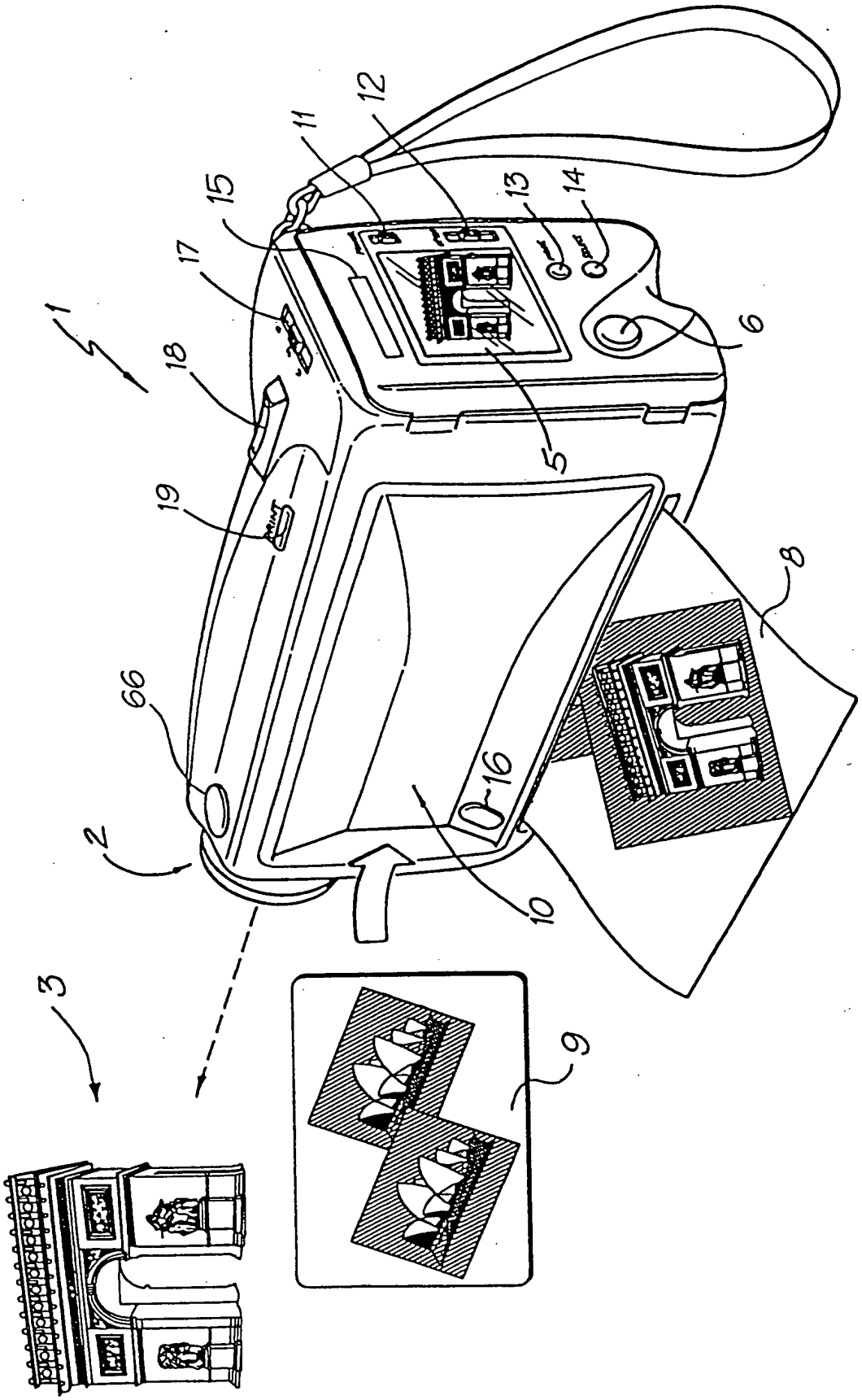


FIG. 1

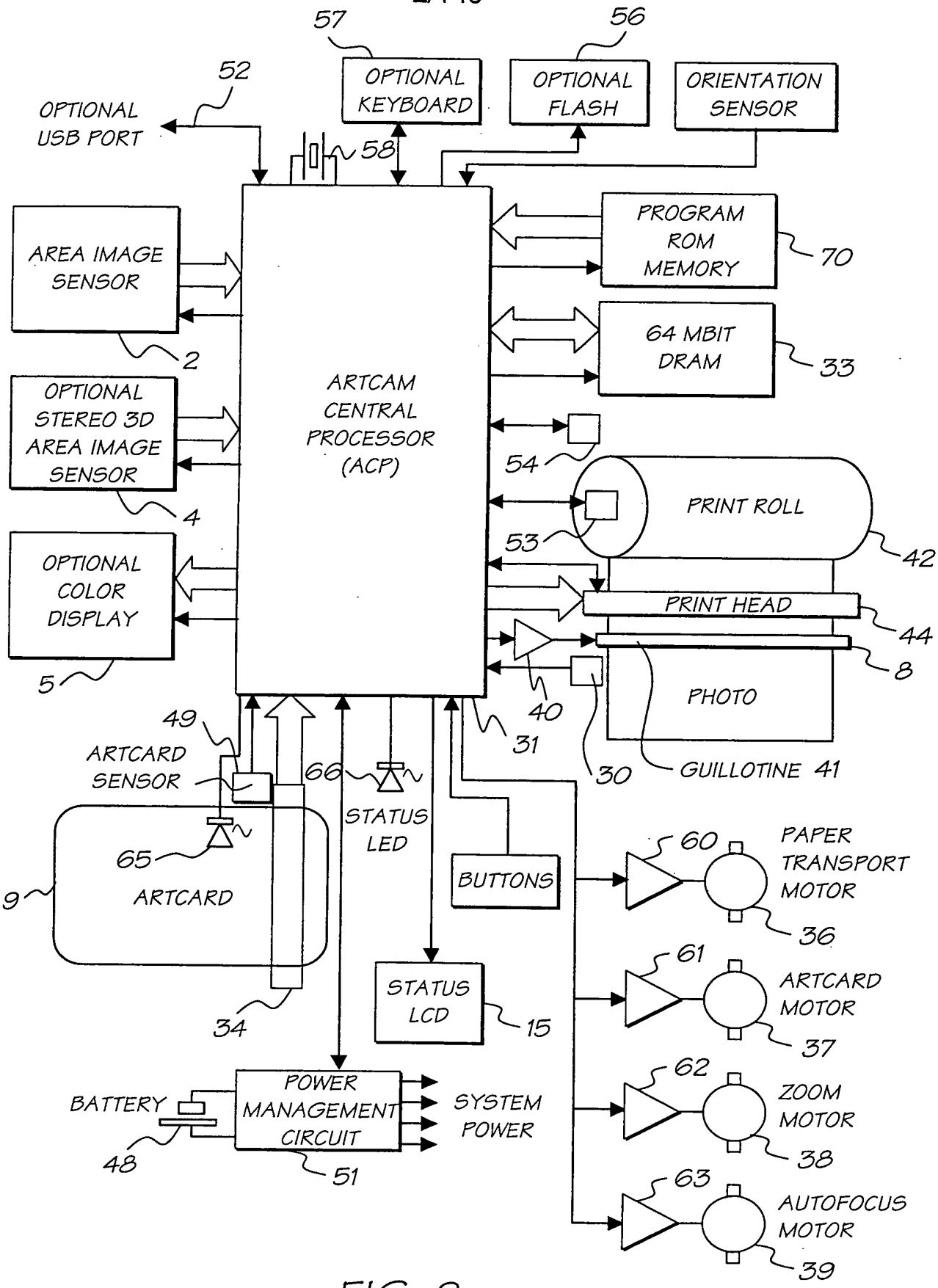

57
OPTIONAL KEYBOARD
56
OPTIONAL FLASH
ORIENTATION SENSOR
52
OPTIONAL USB PORT
58
AREA IMAGE SENSOR
2
PROGRAM ROM MEMORY
70
64 MBIT DRAM
33
ARTCAM CENTRAL PROCESSOR (ACP)
OPTIONAL STEREO 3D AREA IMAGE SENSOR
4
54
PRINT ROLL
42
53
OPTIONAL COLOR DISPLAY
5
PRINT HEAD
44
8
40
PHOTO
49
31
30
GUILLOTINE 41
ARTCARD SENSOR
66
STATUS LED
65
ARTCARD
9
BUTTONS
60
PAPER TRANSPORT MOTOR
36
61
ARTCARD MOTOR
37
STATUS LCD
15
34
62
ZOOM MOTOR
38
BATTERY
48
POWER MANAGEMENT CIRCUIT
51
SYSTEM POWER
63
AUTOFOCUS MOTOR
39

FIG. 2

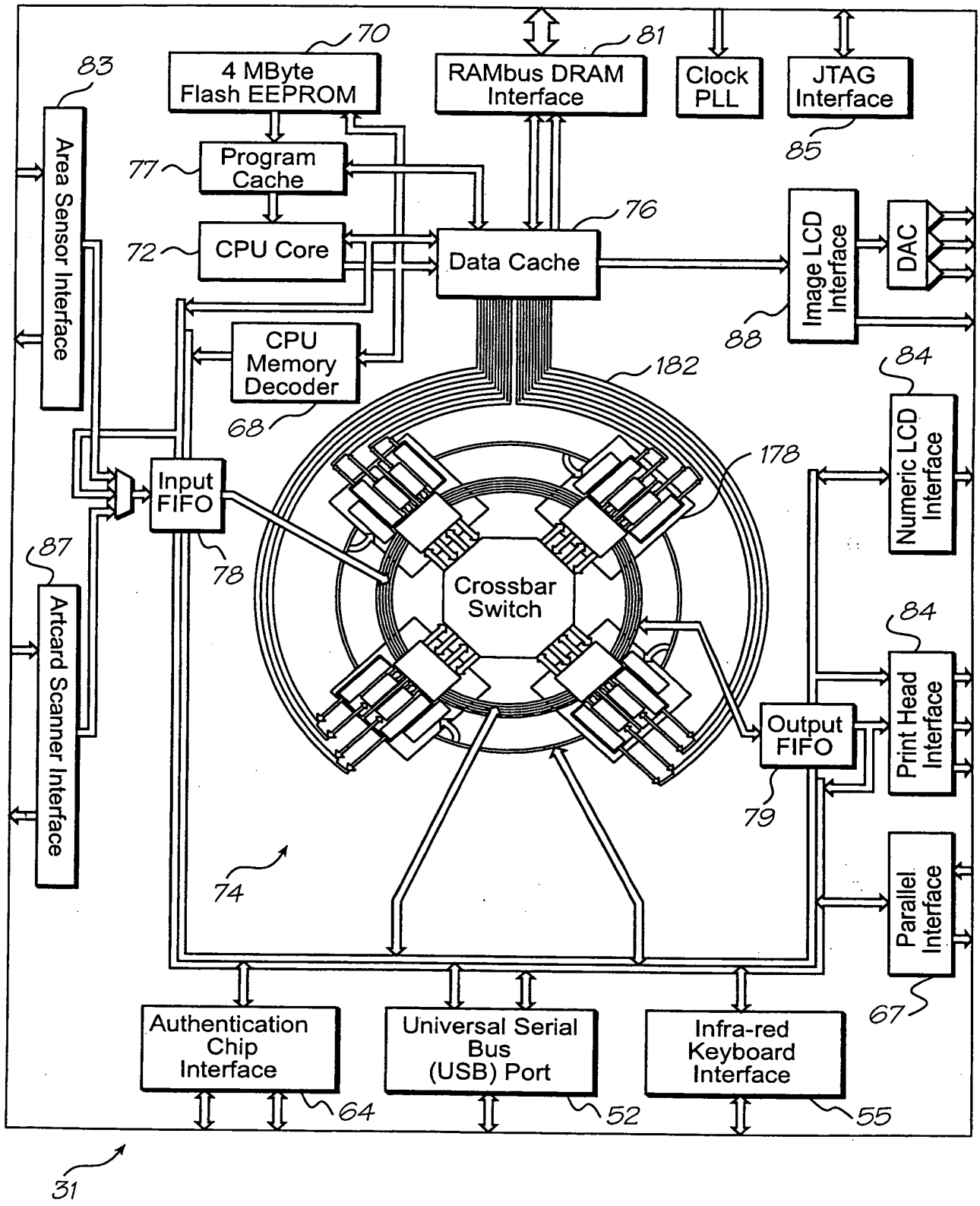

70
4 MByte Flash EEPROM
81
RAMbus DRAM Interface
Clock PLL
JTAG Interface
85
83
Area Sensor Interface
77
Program Cache
72
CPU Core
76
Data Cache
Image LCD Interface
DAC
88
CPU Memory Decoder
68
182
84
Numeric LCD Interface
178
Input FIFO
78
87
Artcard Scanner Interface
Crossbar Switch
84
Print Head Interface
Output FIFO
79
74
Parallel Interface
67
Authentication Chip Interface
64
Universal Serial Bus (USB) Port
52
Infra-red Keyboard Interface
55
31

FIG. 3

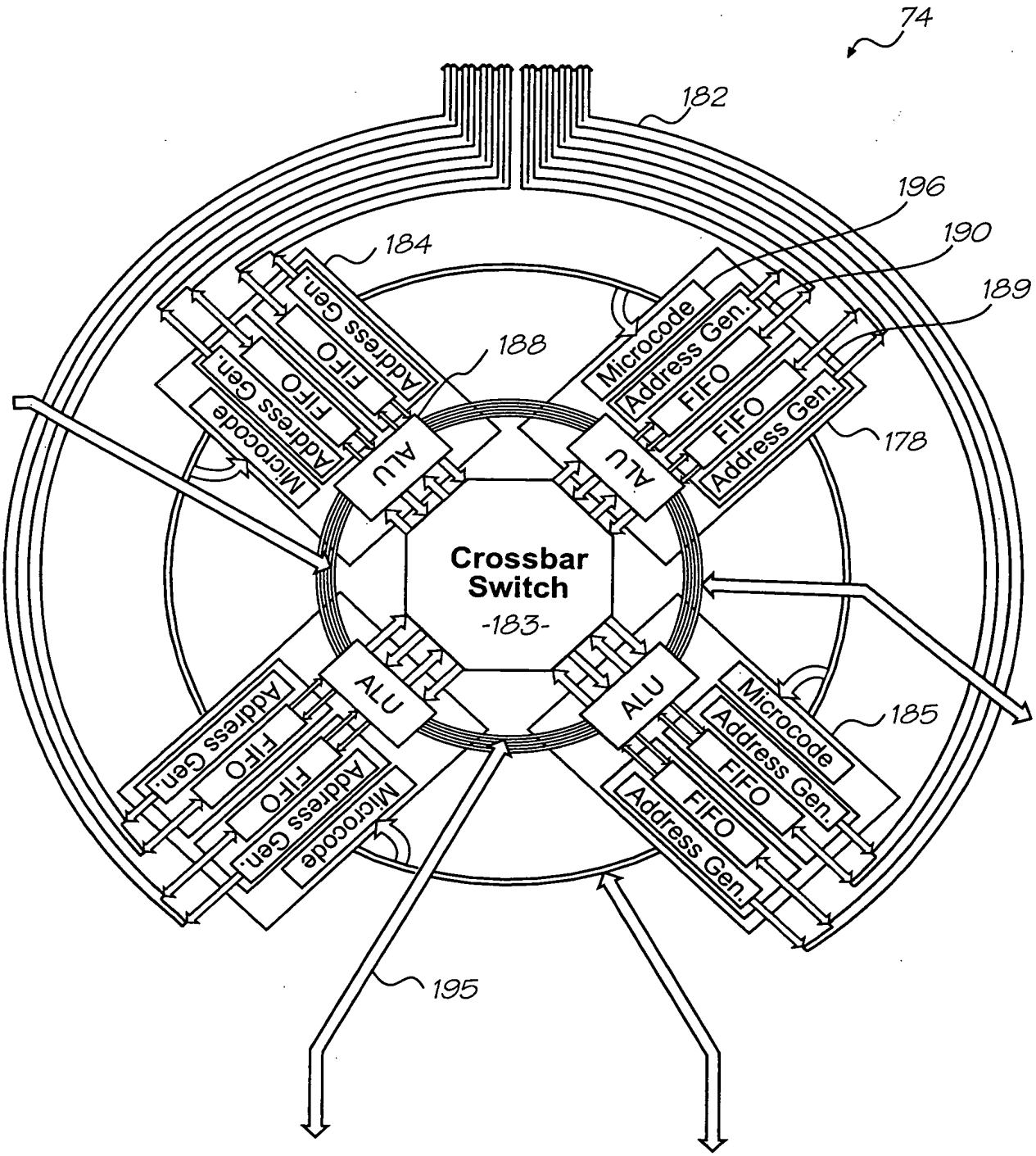

74
182
196
190
189
184
188
178
185
195
Crossbar Switch
-183-
ALU
Microcode
Address Gen.
FIFO
Address Gen.
FIFO
Microcode
Address Gen.
FIFO
Address Gen.
FIFO
Microcode
Address Gen.
FIFO
Address Gen.
FIFO
Microcode
Address Gen.
FIFO
Address Gen.

FIG. 3(a)

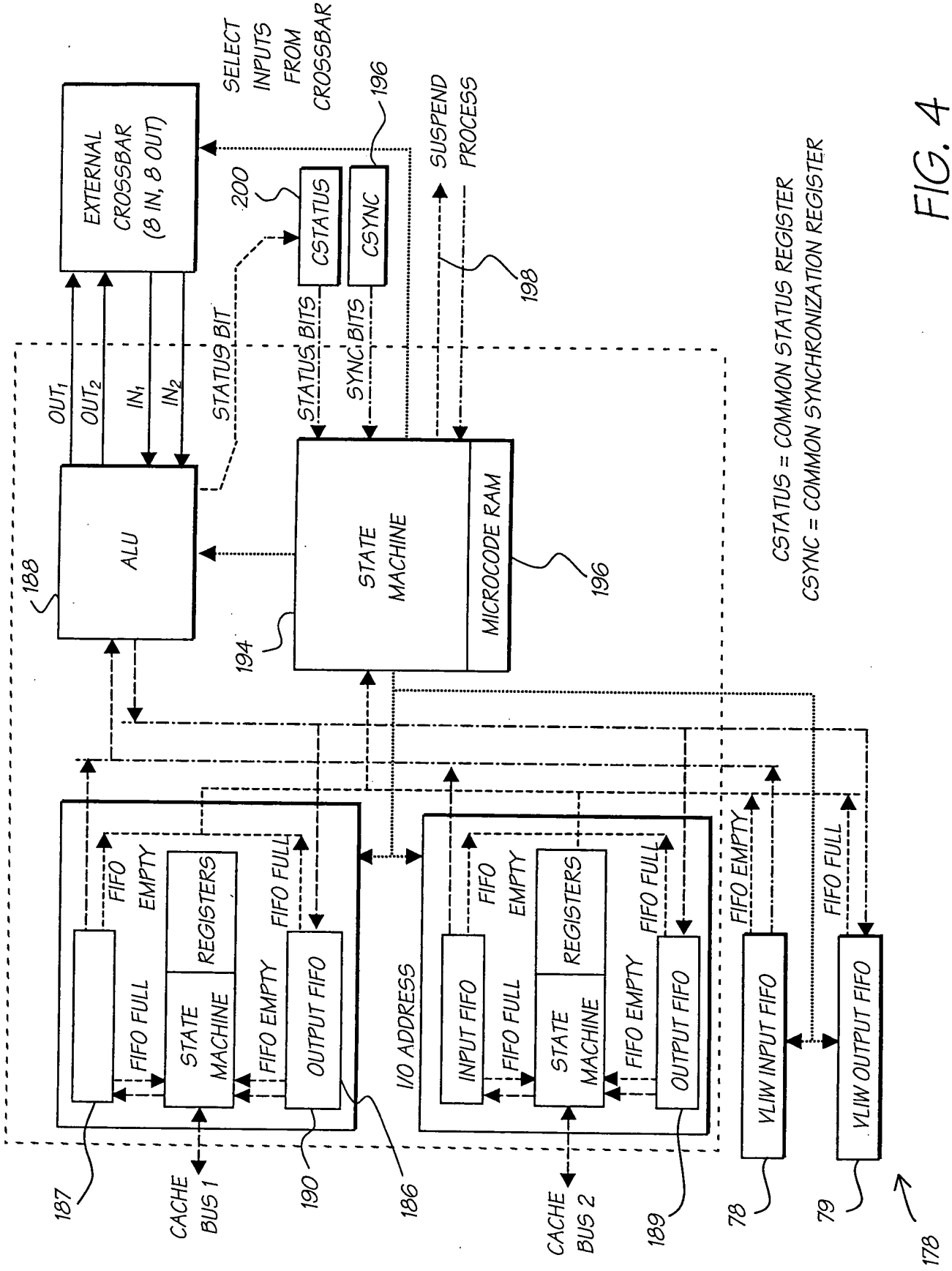

EXTERNAL CROSSBAR (8 IN, 8 OUT)
SELECT INPUTS FROM CROSSBAR
196
SUSPEND PROCESS
200
CSTATUS
CSYNC
198
OUT1
OUT2
IN1
IN2
STATUS BIT
STATUS BITS
SYNC BITS
ALU
188
STATE MACHINE
MICROCODE RAM
196
194
CSTATUS = COMMON STATUS REGISTER
CSYNC = COMMON SYNCHRONIZATION REGISTER
FIFO EMPTY
REGISTERS
FIFO FULL
FIFO FULL
STATE MACHINE
FIFO EMPTY
OUTPUT FIFO
187
CACHE BUS 1
190
186
I/O ADDRESS
INPUT FIFO
FIFO EMPTY
REGISTERS
FIFO FULL
FIFO FULL
STATE MACHINE
FIFO EMPTY
OUTPUT FIFO
CACHE BUS 2
189
FIFO EMPTY
VLIW INPUT FIFO
78
FIFO FULL
VLIW OUTPUT FIFO
79
178

FIG. 4

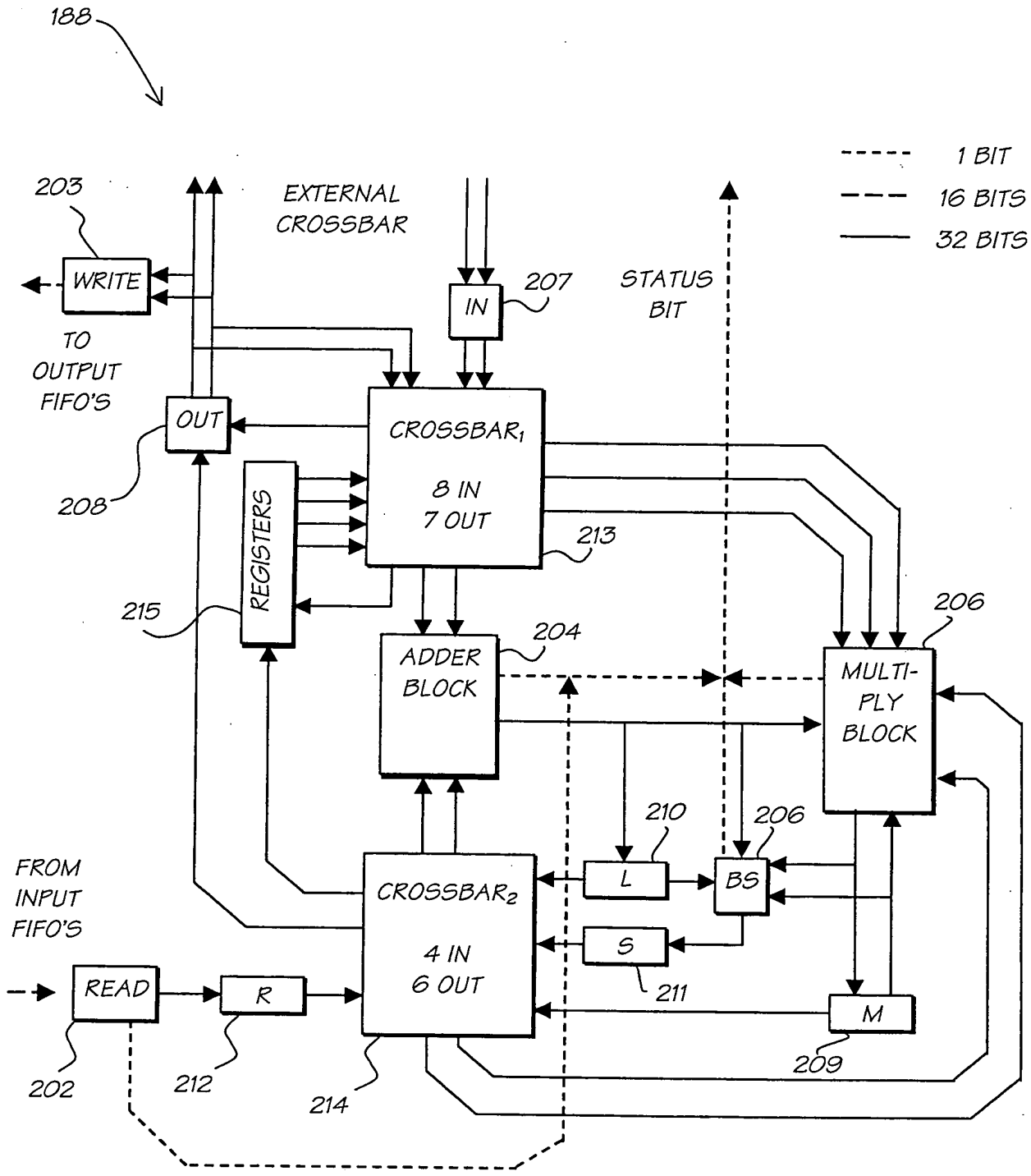

188
1 BIT
16 BITS
32 BITS
203
EXTERNAL CROSSBAR
WRITE
TO OUTPUT FIFO'S
IN
207
STATUS BIT
OUT
208
CROSSBAR1
8 IN
7 OUT
213
REGISTERS
215
ADDER BLOCK
204
MULTI-PLY BLOCK
206
FROM INPUT FIFO'S
CROSSBAR2
4 IN
6 OUT
210
L
BS
S
211
M
209
READ
R
202
212
214

FIG. 5

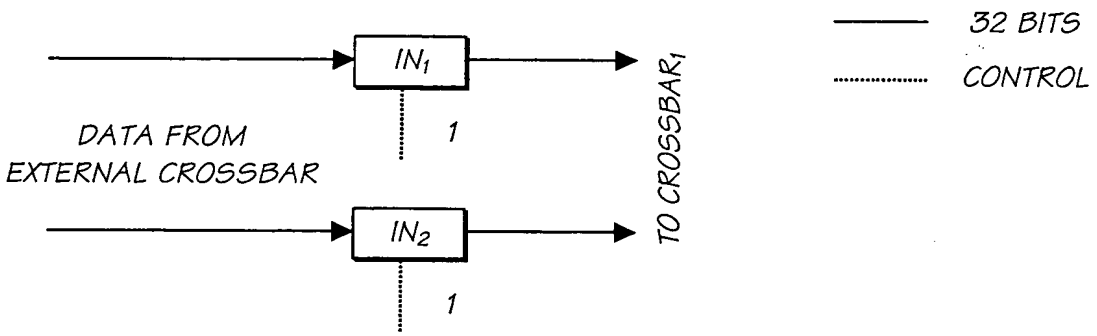

32 BITS
CONTROL
IN1
1
DATA FROM
EXTERNAL CROSSBAR
IN2
1
TO CROSSBAR1

FIG. 6

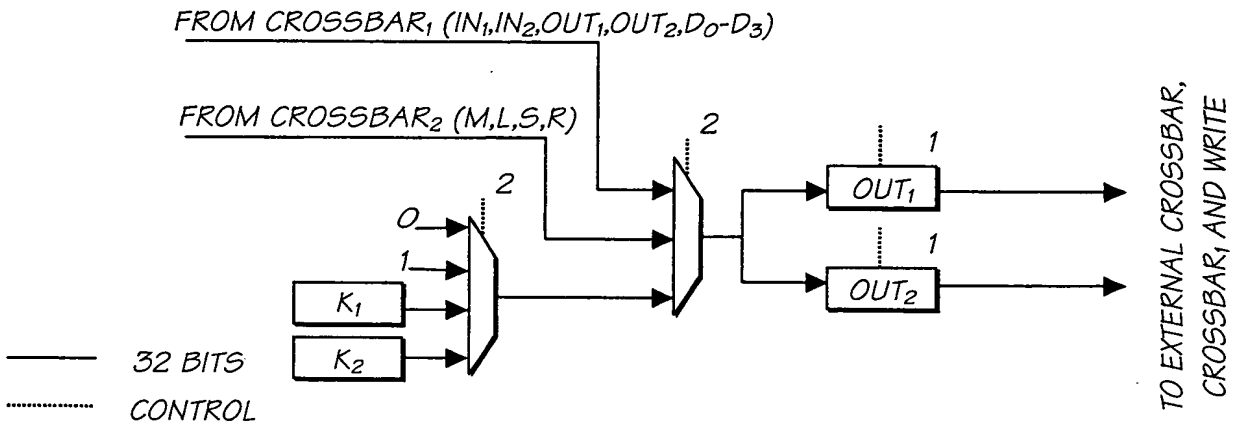

FROM CROSSBAR1 (IN1,IN2,OUT1,OUT2,D0-D3)
FROM CROSSBAR2 (M,L,S,R)
2
2
0
1
K1
K2
1
OUT1
1
OUT2
32 BITS
CONTROL
TO EXTERNAL CROSSBAR,
CROSSBAR1 AND WRITE

FIG. 7

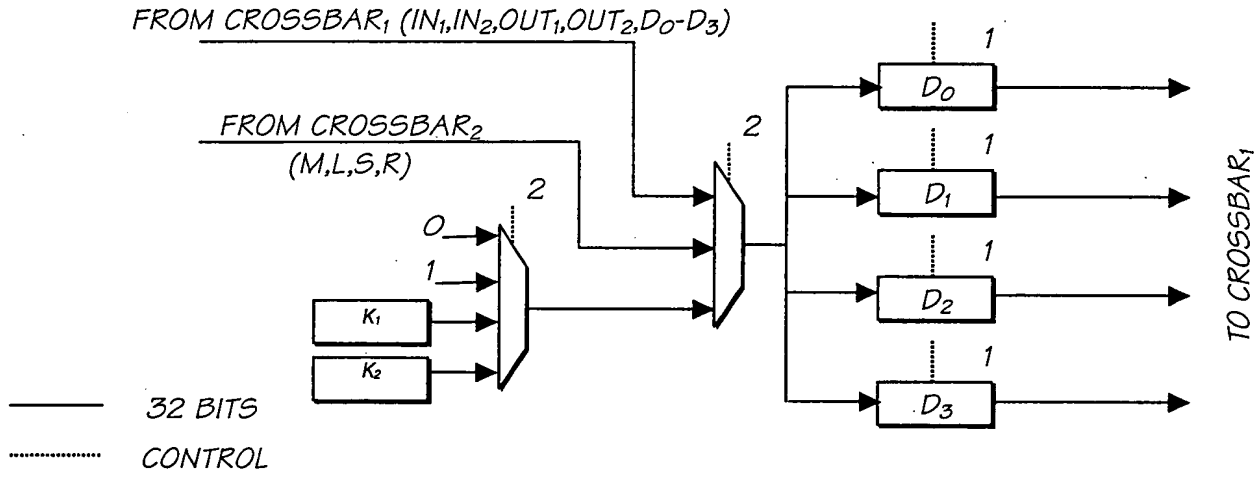

FROM CROSSBAR1 (IN1,IN2,OUT1,OUT2,D0-D3)
FROM CROSSBAR2
(M,L,S,R)
2
0
1
K1
K2
2
1
D0
1
D1
1
D2
1
D3
TO CROSSBAR1
32 BITS
CONTROL

FIG. 8

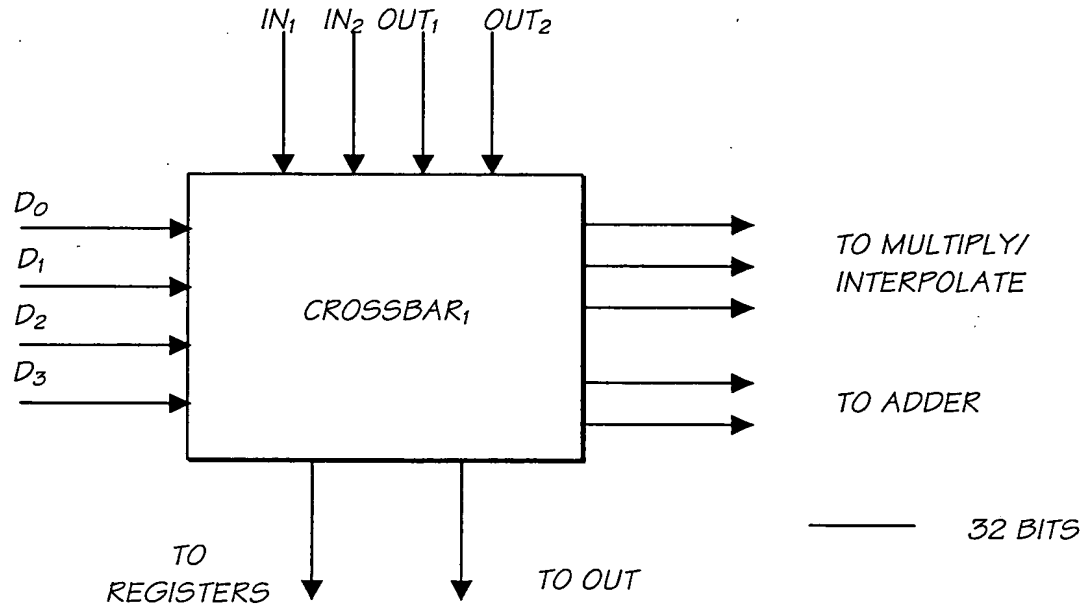

IN1
IN2
OUT1
OUT2
D0
D1
D2
D3
CROSSBAR1
TO MULTIPLY/
INTERPOLATE
TO ADDER
32 BITS
TO
REGISTERS
TO OUT

FIG. 9

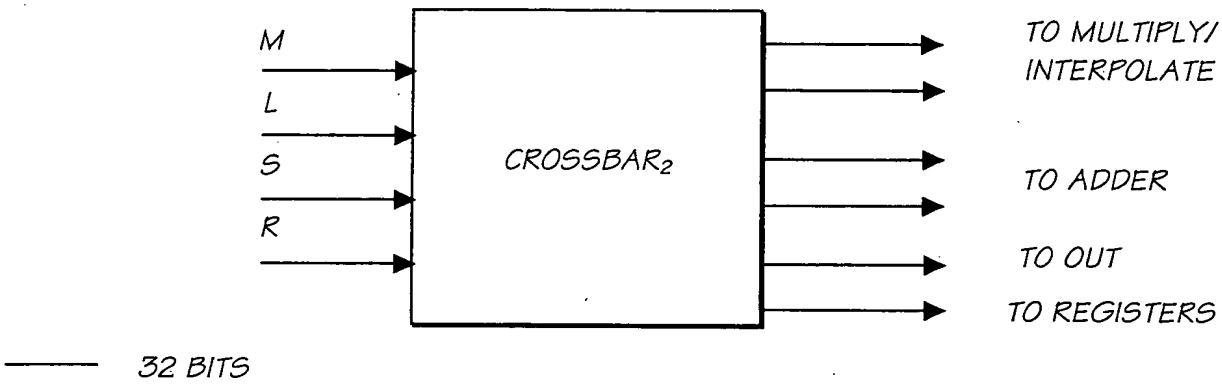

M
L
S
R
CROSSBAR2
TO MULTIPLY/
INTERPOLATE
TO ADDER
TO OUT
TO REGISTERS
32 BITS

FIG. 10

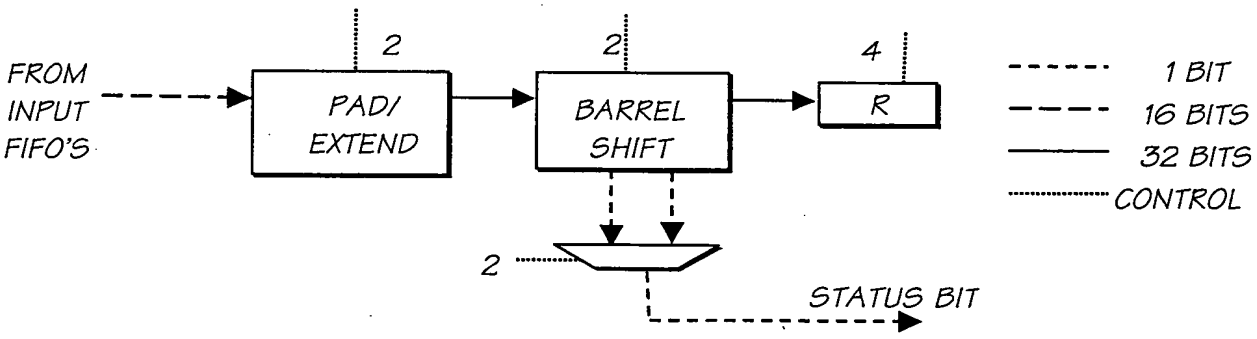

FROM
INPUT
FIFO'S
2
PAD/
EXTEND
2
BARREL
SHIFT
4
R
2
STATUS BIT
1 BIT
16 BITS
32 BITS
CONTROL

FIG. 11

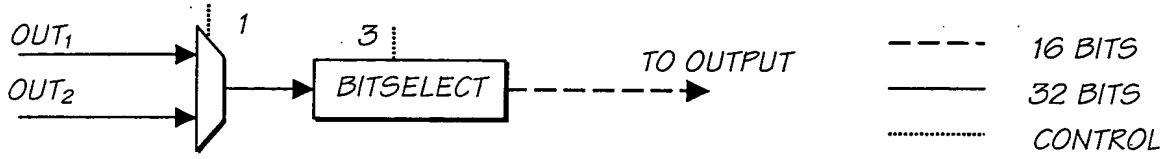


FIG. 12

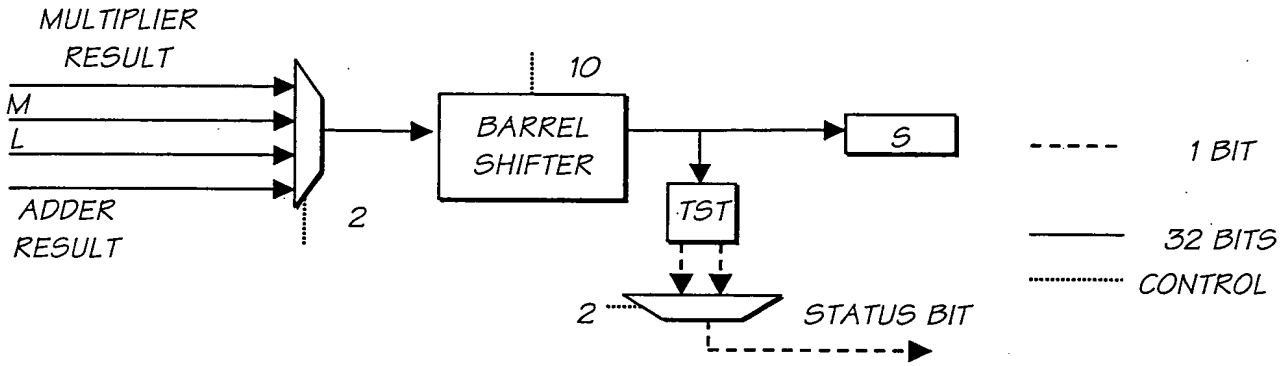


FIG. 13

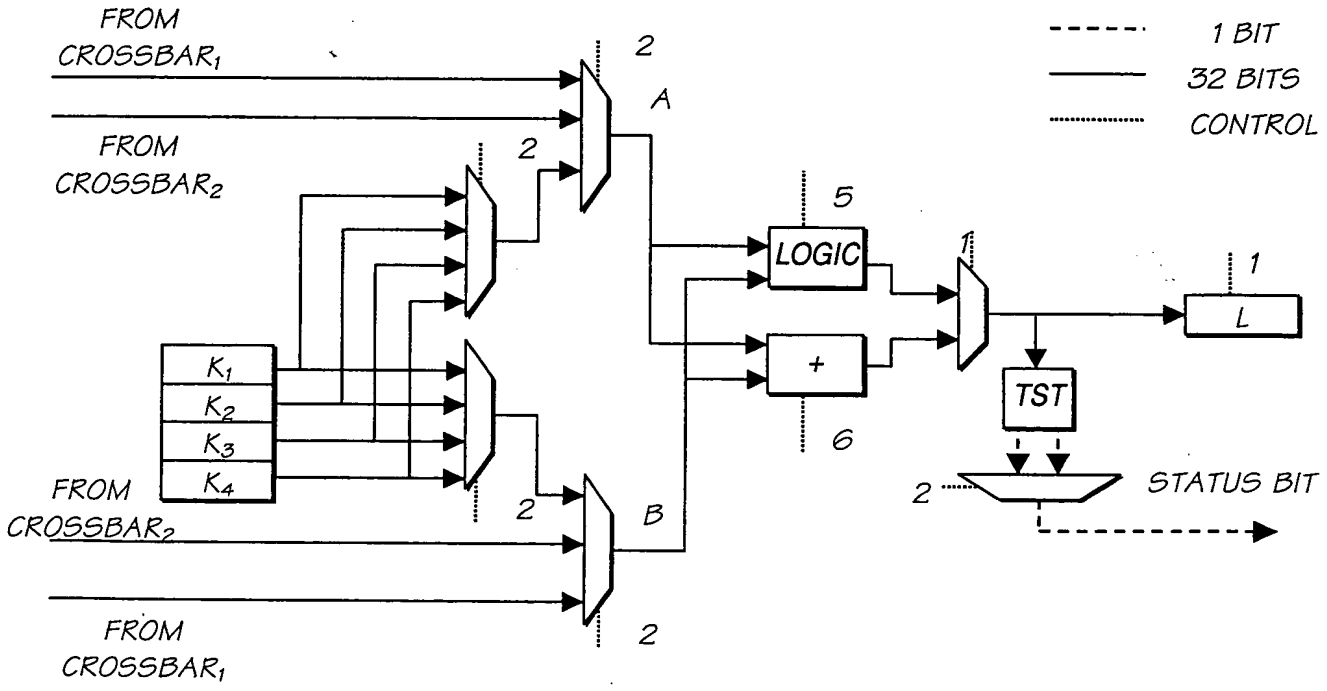


FIG. 14

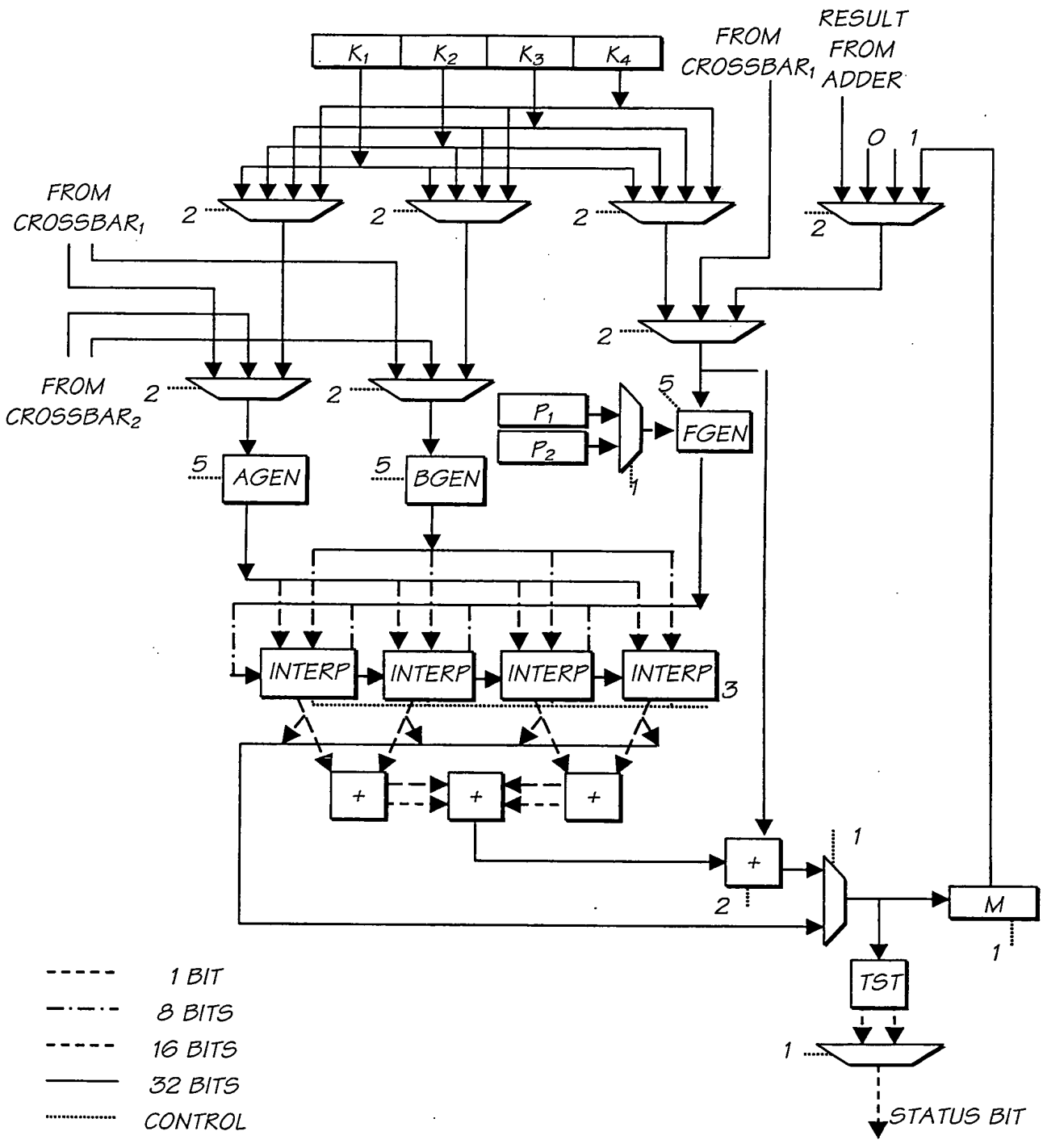

K1
K2
K3
K4
FROM CROSSBAR1
RESULT FROM ADDER
0
1
FROM CROSSBAR1
2
2
2
2
FROM CROSSBAR2
2
2
2
5
P1
P2
FGEN
1
5
AGEN
5
BGEN
INTERP
INTERP
INTERP
INTERP
3
+
+
+
+
2
1
M
1
TST
1
STATUS BIT
1 BIT
8 BITS
16 BITS
32 BITS
CONTROL

FIG. 15

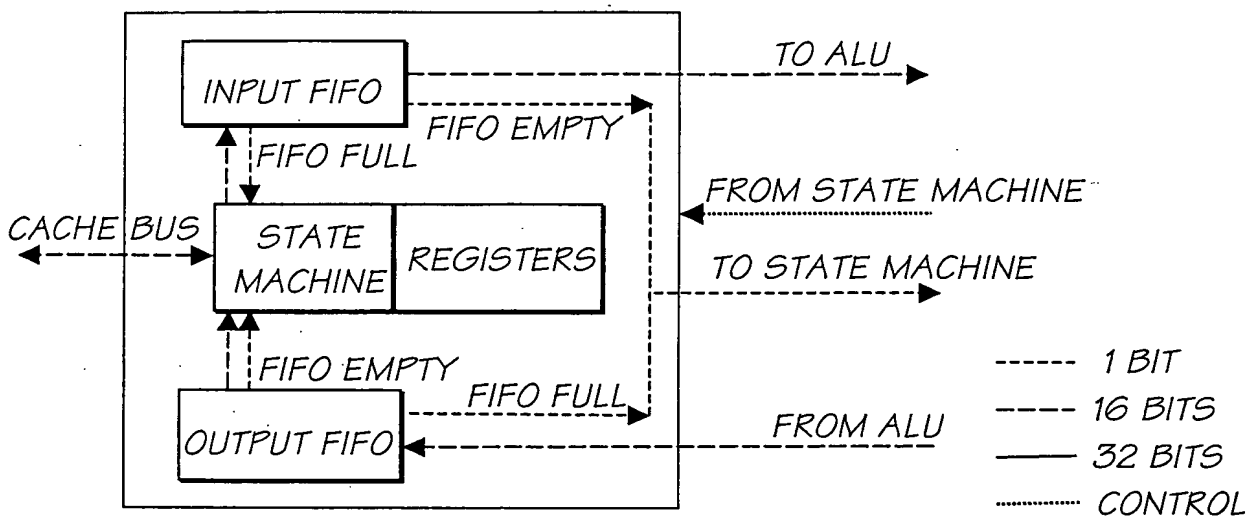


FIG. 16

ORDER OF PIXELS PRESENTED BY A SEQUENTIAL READ ITERATOR ON A 4 X 2 IMAGE WITH PADDING.

0	1	2	3	
4	5	6	7	

FIG. 17

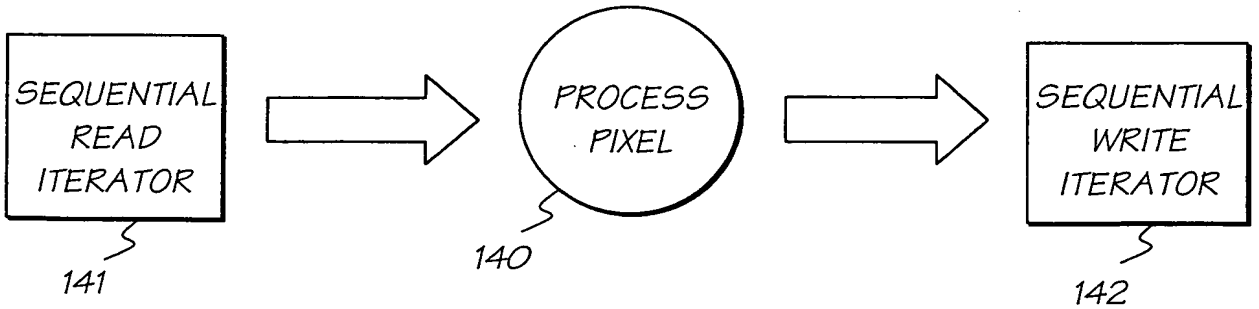
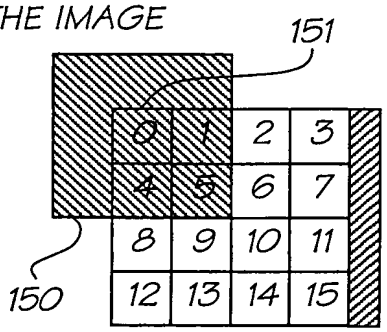


FIG. 18

A 3x3 BOX VIEW TRAVERSES THE PIXELS IN ORDER: 0, 1, 2, 3, 4, 5, 6, 7, 8 ETC, PLACING A 3x3 BOX CENTERED OVER EACH PIXEL...

3x3 BOX VIEW OF FIRST PIXEL IN IMAGE = 9 PIXELS, 5 OF WHICH ARE OUTSIDE THE IMAGE



FIRST 9 PIXELS FROM THE BOX READ ITERATOR:

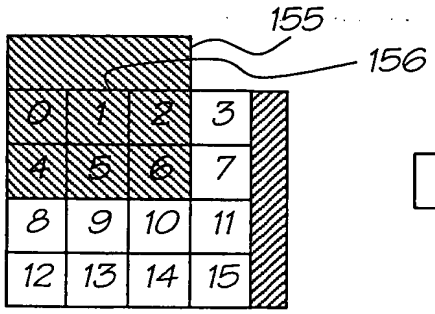
152

IF DUPLICATION OF EDGE PIXELS IS ON: 0, 0, 0, 0, 0, 1, 4, 4, 5

IF DUPLICATION OF EDGE PIXELS IS OFF: V, V, V, V, 0, 1, V, 4, 5 WHERE V IS CONSTANTPIXEL REGISTER VALUE REPRESENTING "OUTSIDE THE IMAGE"

153

3x3 BOX VIEW OF SECOND PIXEL IN IMAGE = 9 PIXELS, 3 OF WHICH ARE OUTSIDE THE IMAGE



SECOND 9 PIXELS FROM THE BOX READ ITERATOR:

IF DUPLICATION OF EDGE PIXELS IS ON: 0, 1, 2, 0, 1, 2, 4, 5, 6

IF DUPLICATION OF EDGE PIXELS IS OFF: V, V, V, 0, 1, 2, 4, 5, 6 WHERE V IS CONSTANTPIXEL REGISTER VALUE REPRESENTING "OUTSIDE THE IMAGE"

FIG. 19

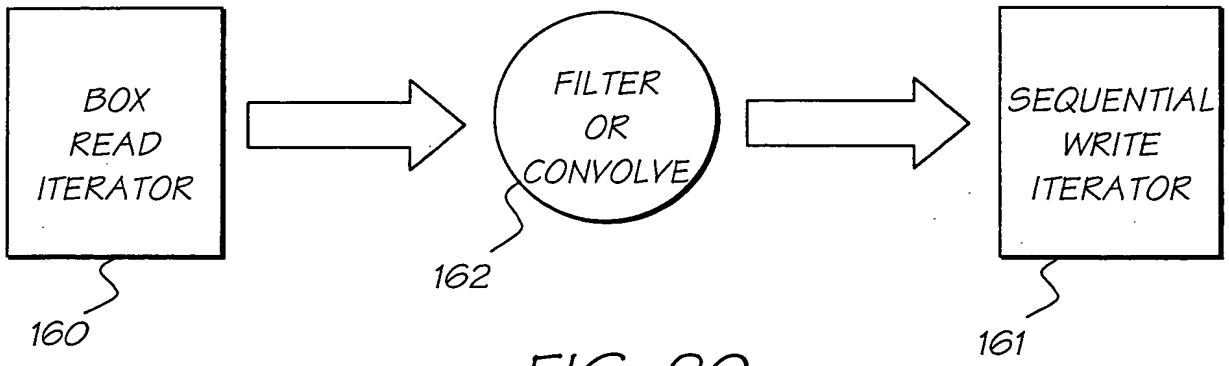


FIG. 20

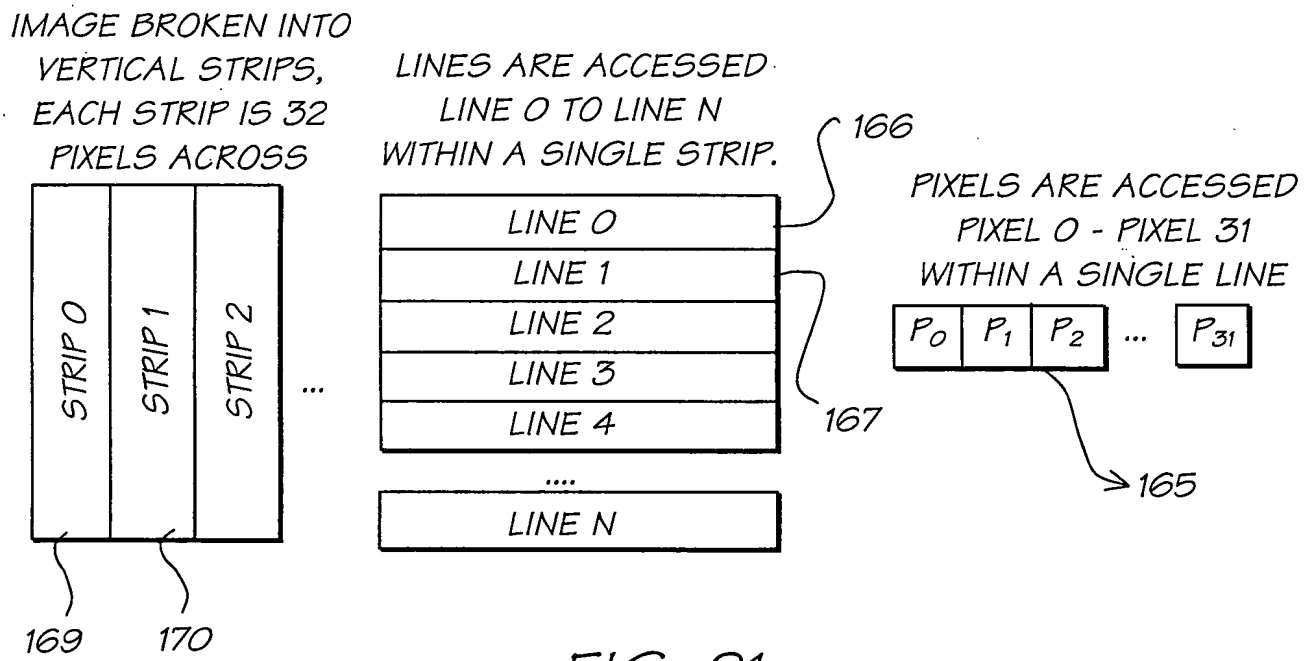


FIG. 21

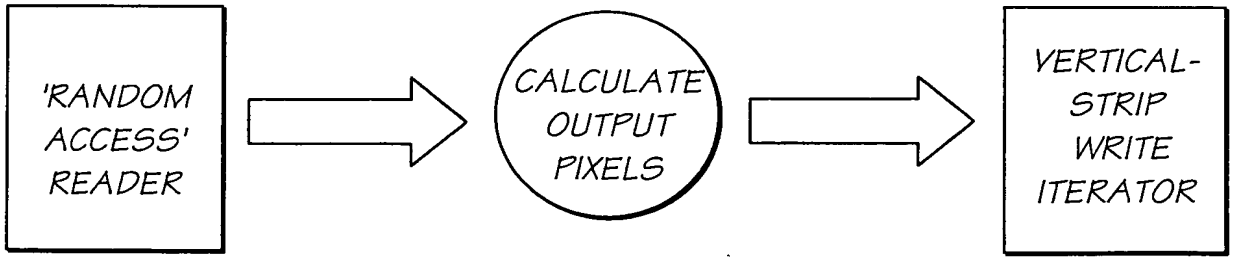


FIG. 22

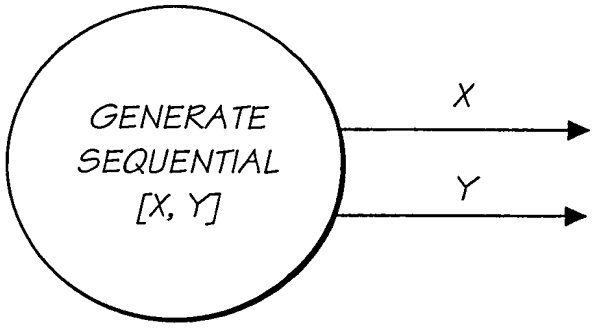

GENERATE
SEQUENTIAL
[X, Y]
X
Y

FIG. 23

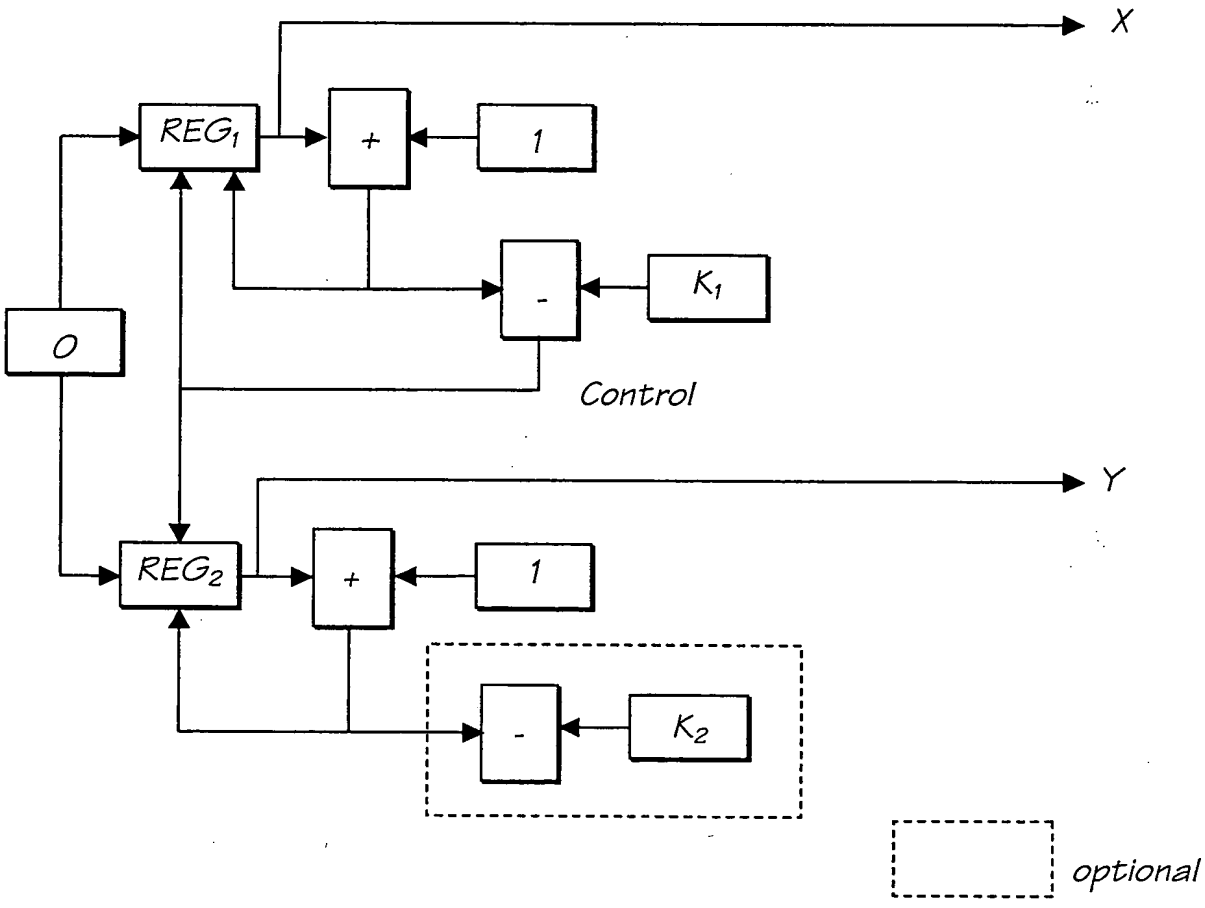

X
REG1
+
1
-
K1
0
Control
Y
REG2
+
1
-
K2
optional

FIG. 24

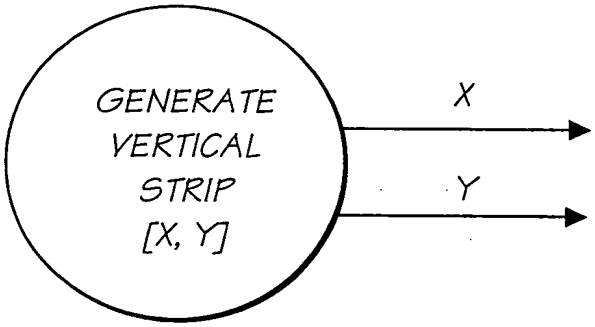

GENERATE
VERTICAL
STRIP
[X, Y]
X
Y

FIG. 25

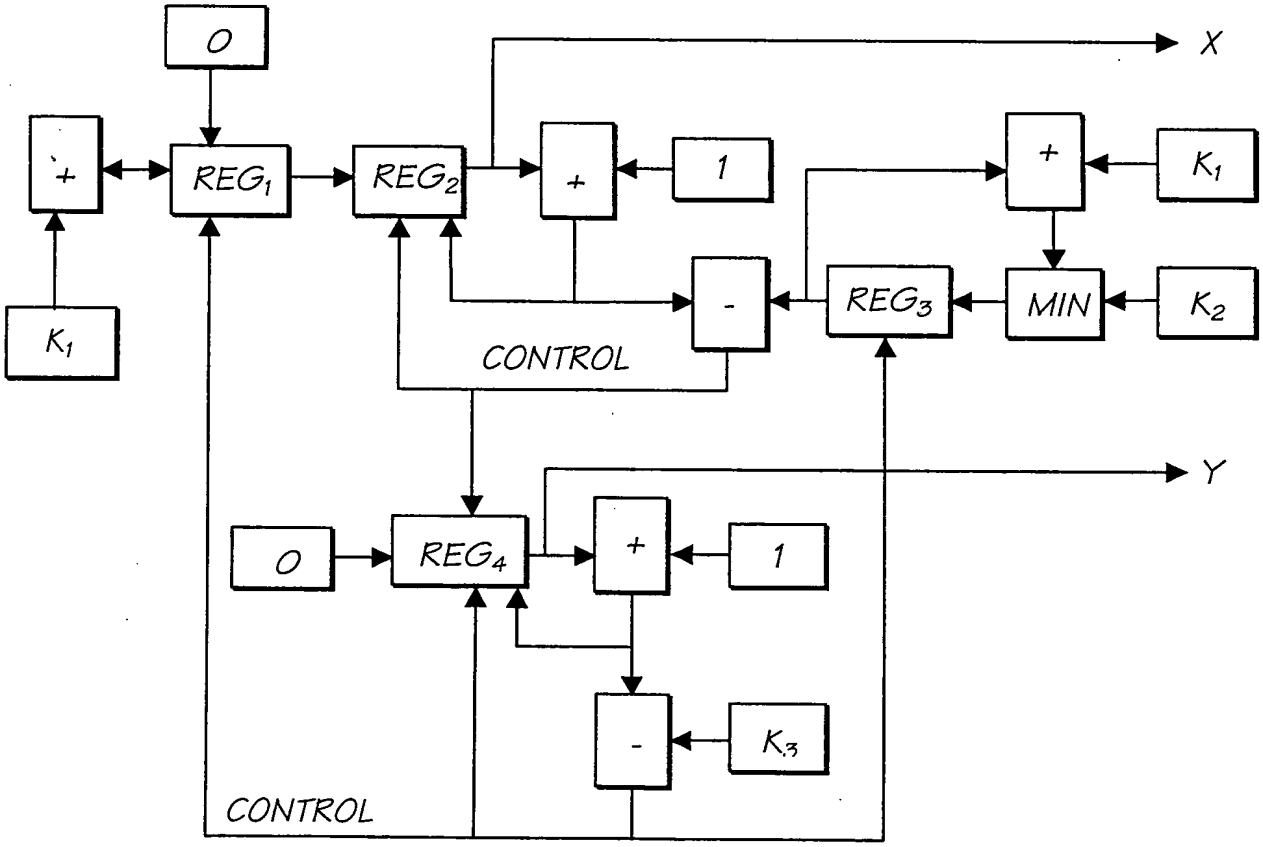
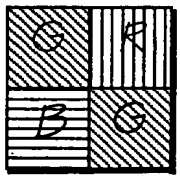

0
X
+
REG1
REG2
+
1
+
K1
K1
-
REG3
MIN
K2
CONTROL
Y
0
REG4
+
1
-
K3
CONTROL

FIG. 26



2X2 PIXEL BLOCK FROM SENSOR

FIG. 27

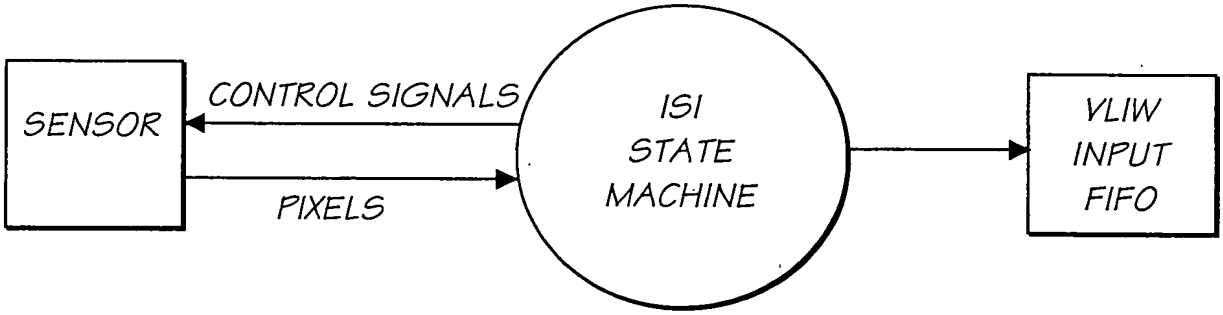


FIG. 28

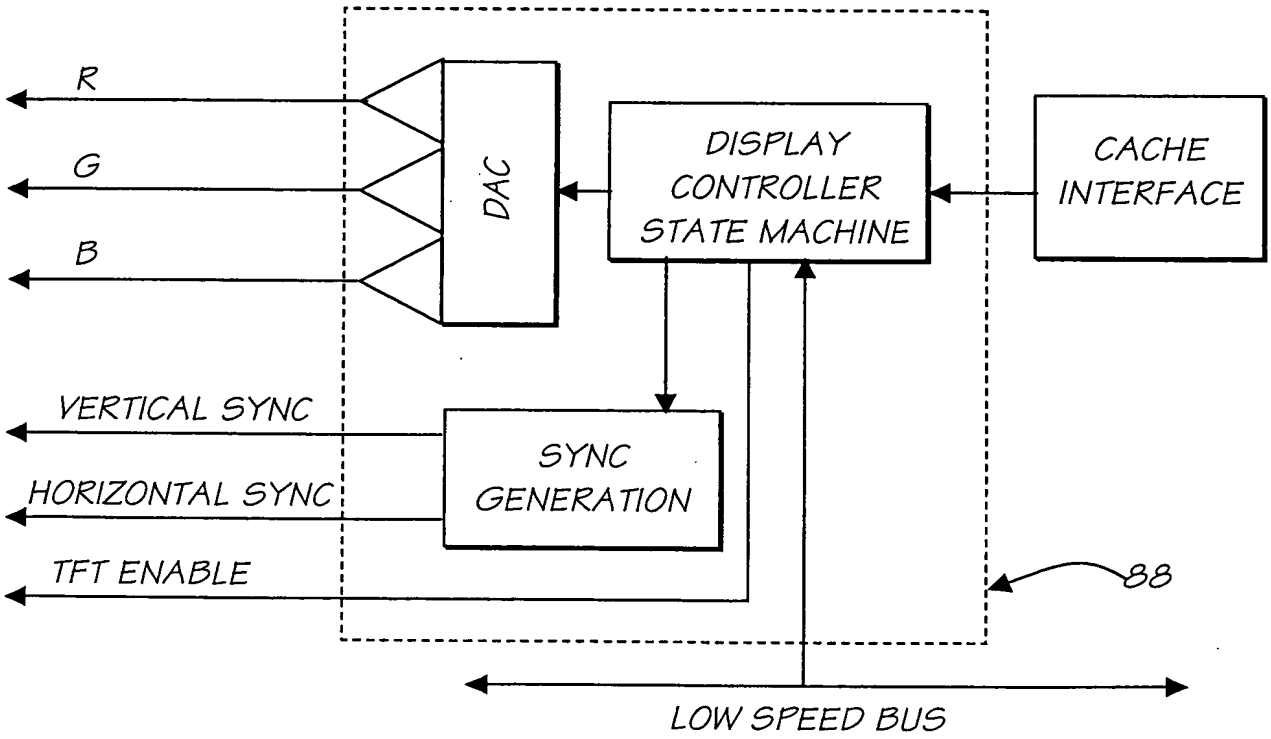
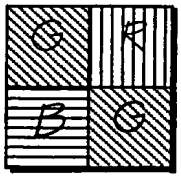


FIG. 29



2X2 PIXEL BLOCK FROM CCD

FIG. 30

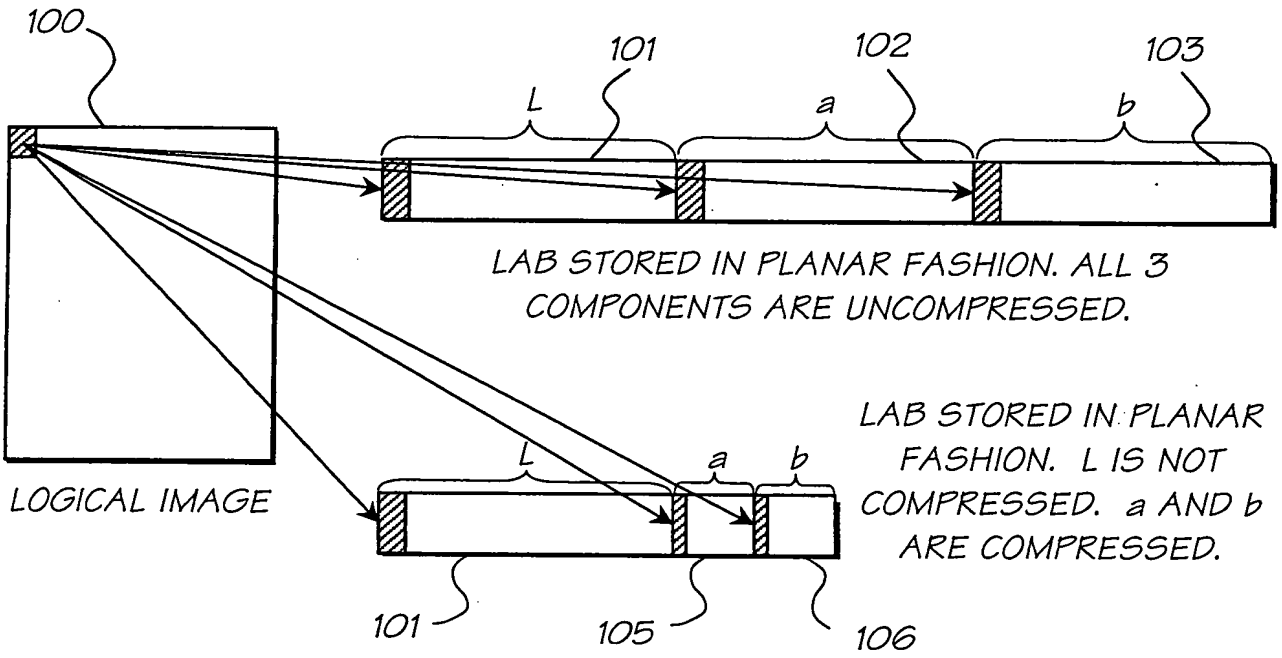


FIG. 31

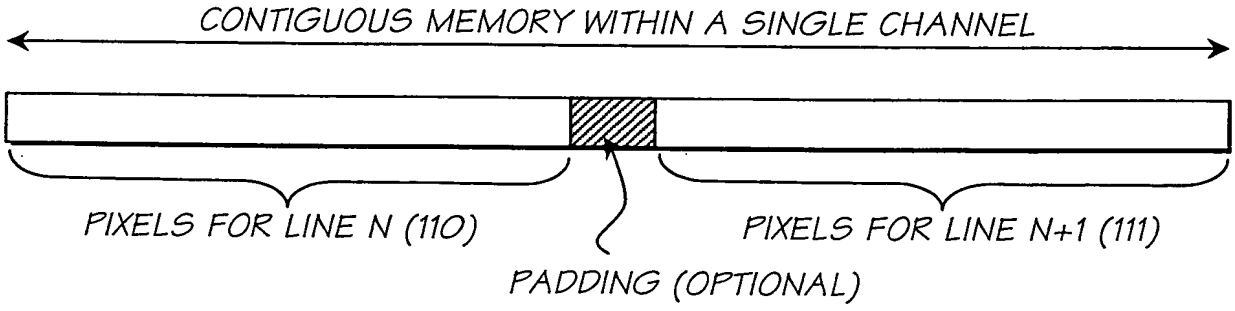


FIG. 32

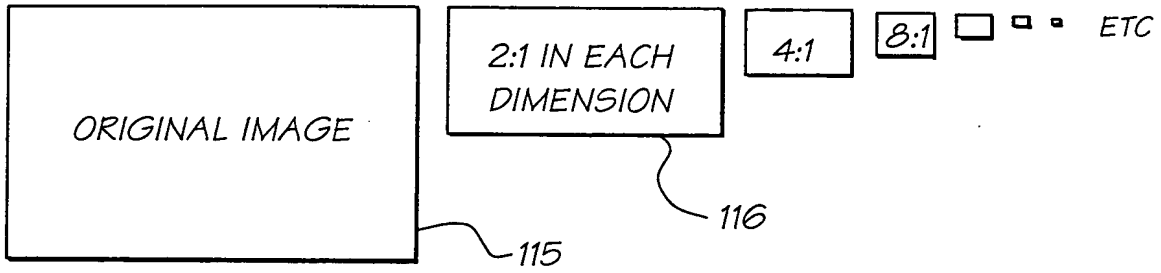


FIG. 33

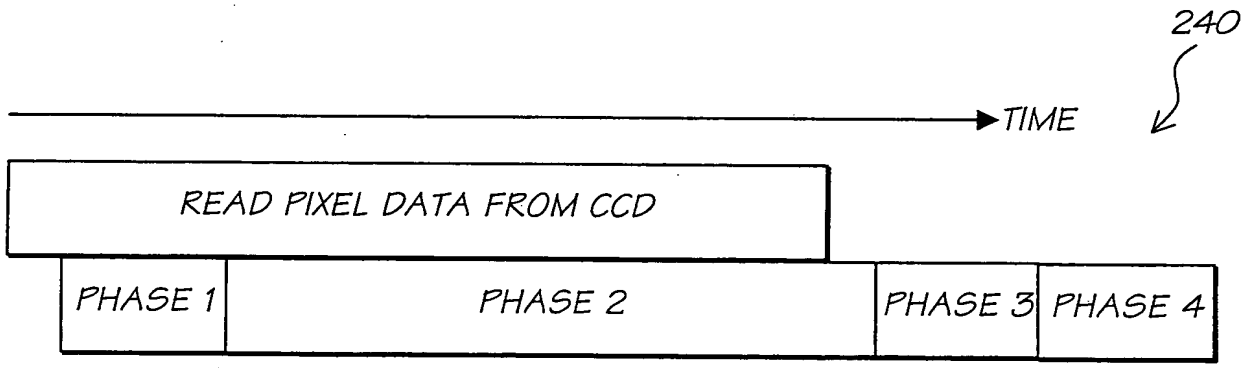


FIG. 34

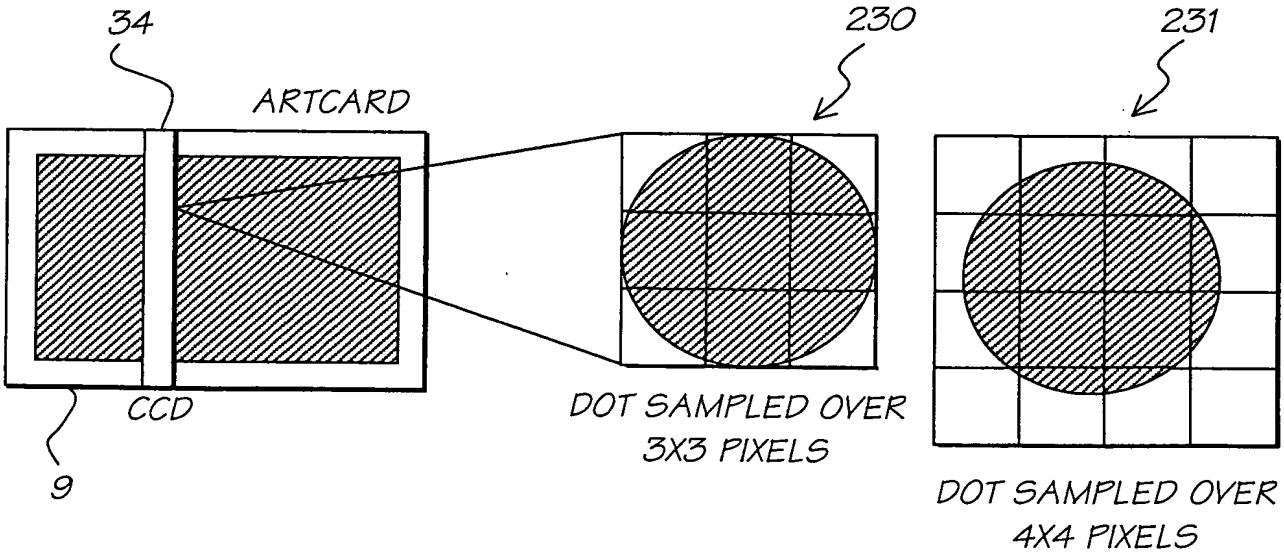


FIG. 35

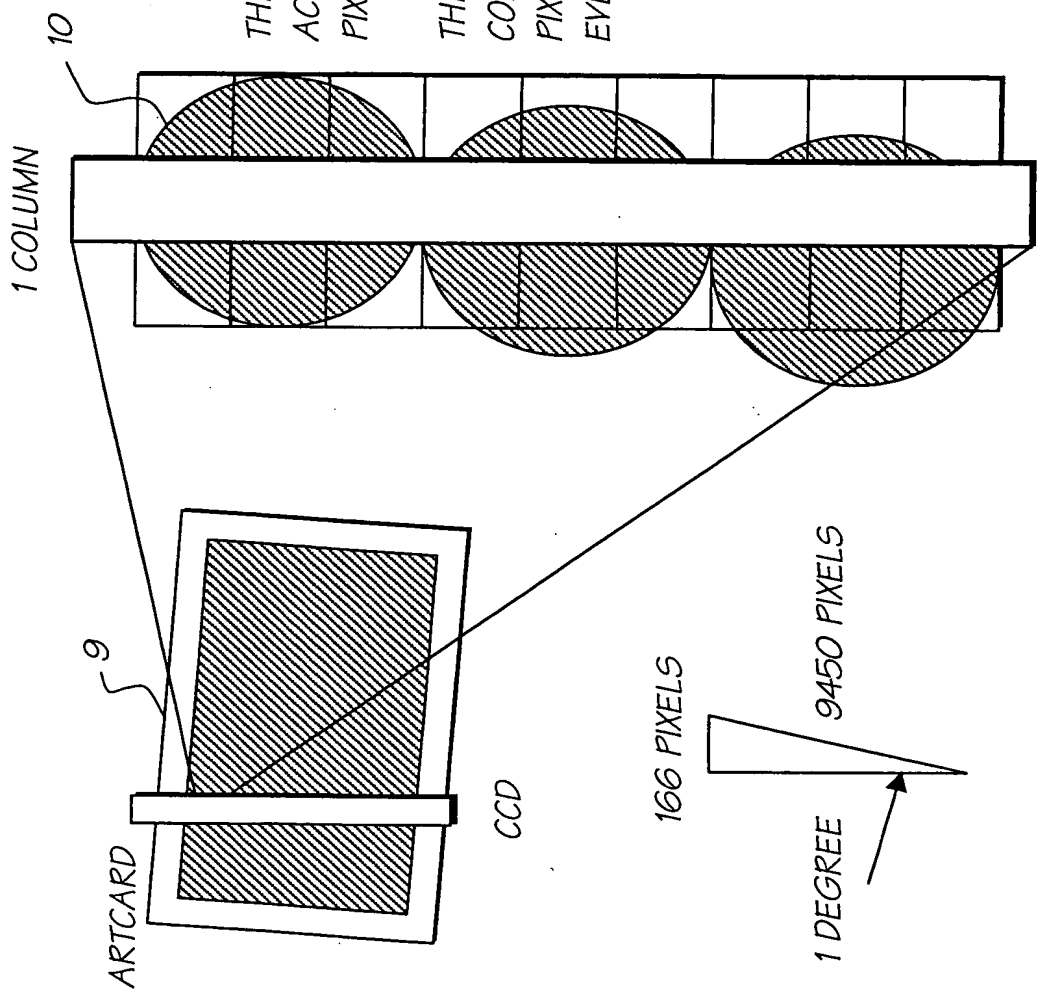

THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).
1 COLUMN
10
9
ARTCARD
CCD
166 PIXELS
9450 PIXELS
1 DEGREE

FIG. 36

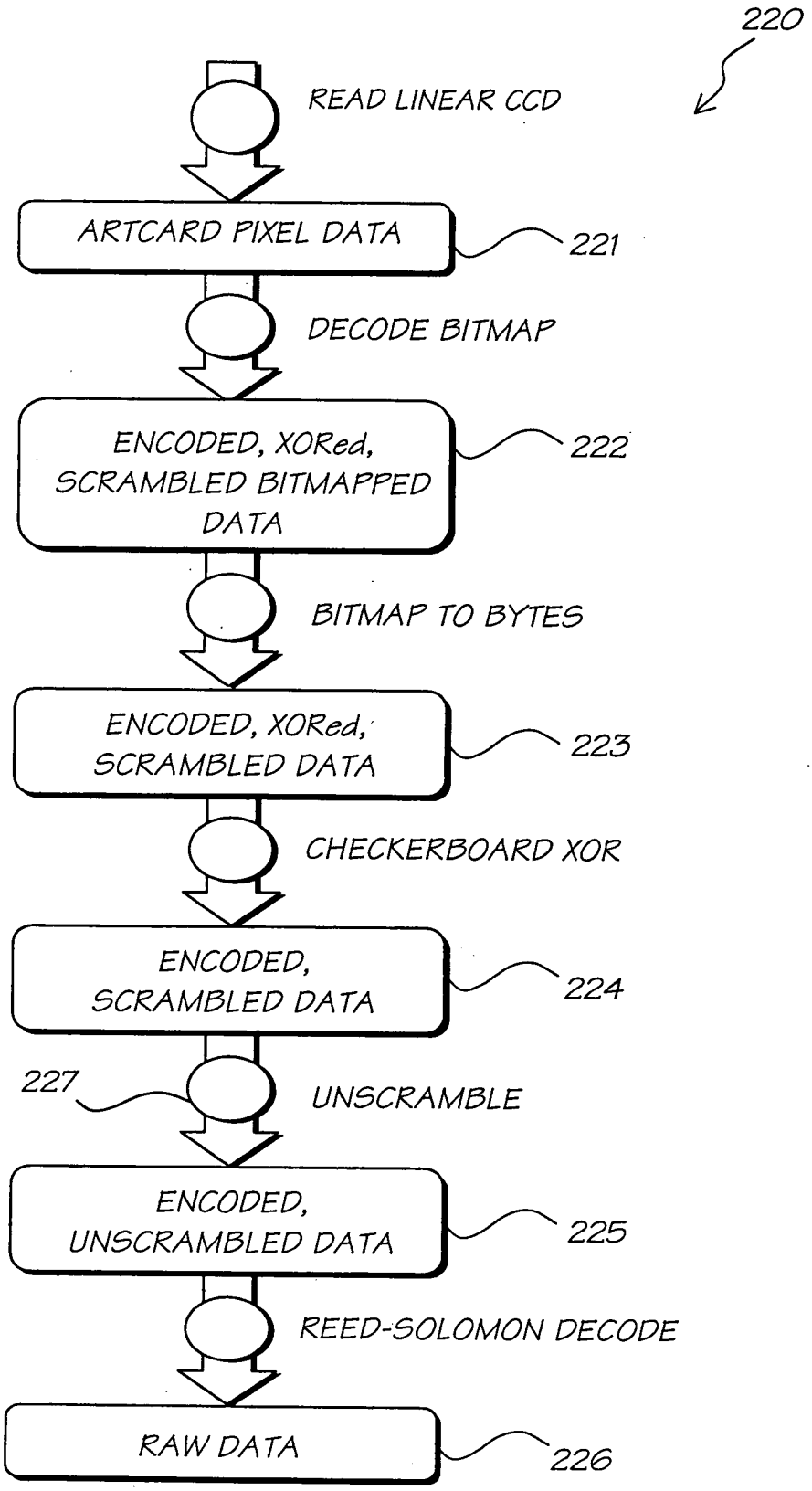

220
READ LINEAR CCD
ARTCARD PIXEL DATA
221
DECODE BITMAP
ENCODED, XORed, SCRAMBLED BITMAPPED DATA
222
BITMAP TO BYTES
ENCODED, XORed, SCRAMBLED DATA
223
CHECKERBOARD XOR
ENCODED, SCRAMBLED DATA
224
227
UNSCRAMBLE
ENCODED, UNSCRAMBLED DATA
225
REED-SOLOMON DECODE
RAW DATA
226

FIG. 37

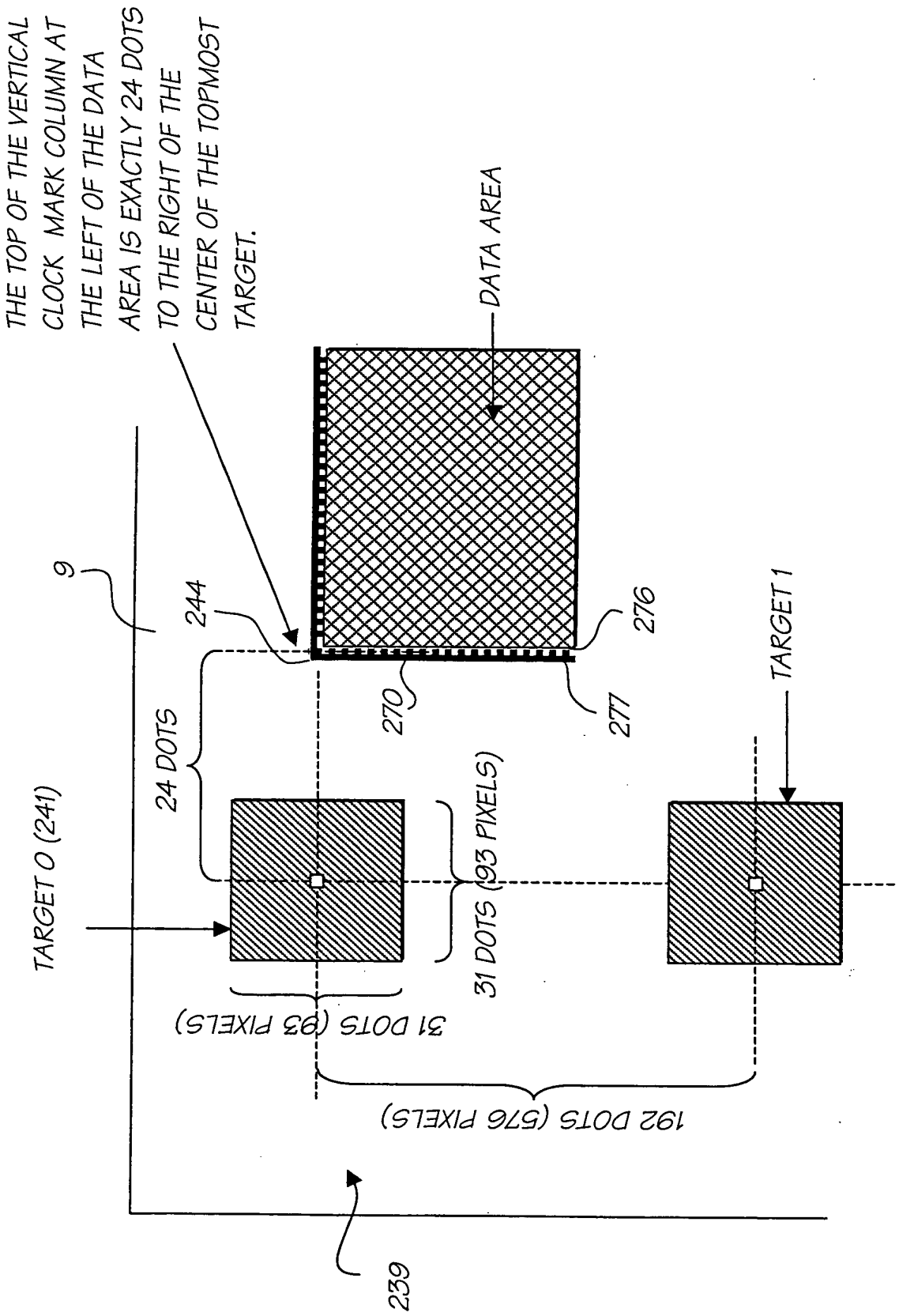

THE TOP OF THE VERTICAL CLOCK MARK COLUMN AT THE LEFT OF THE DATA AREA IS EXACTLY 24 DOTS TO THE RIGHT OF THE CENTER OF THE TOPMOST TARGET.
DATA AREA
9
244
276
270
277
24 DOTS
TARGET 0 (241)
TARGET 1
31 DOTS (93 PIXELS)
31 DOTS (93 PIXELS)
192 DOTS (576 PIXELS)
239

FIG. 38

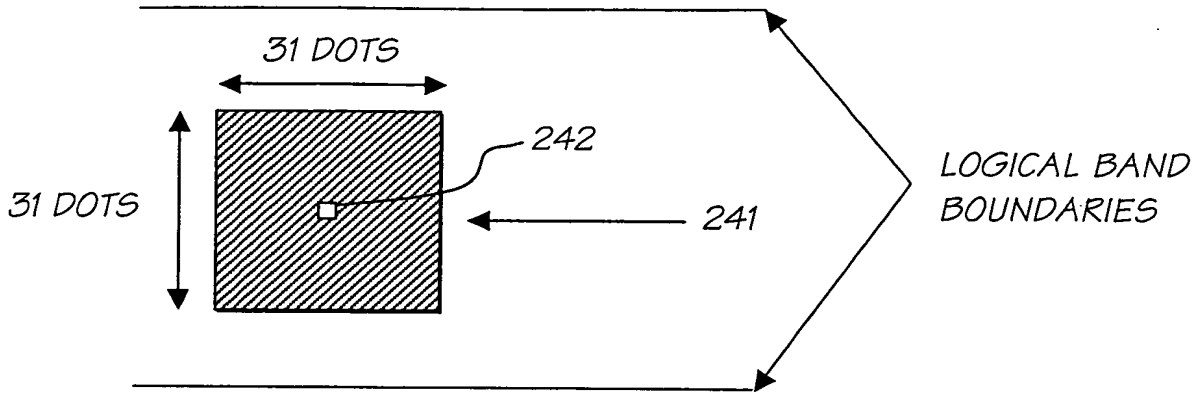

31 DOTS
31 DOTS
242
241
LOGICAL BAND
BOUNDARIES

FIG. 39

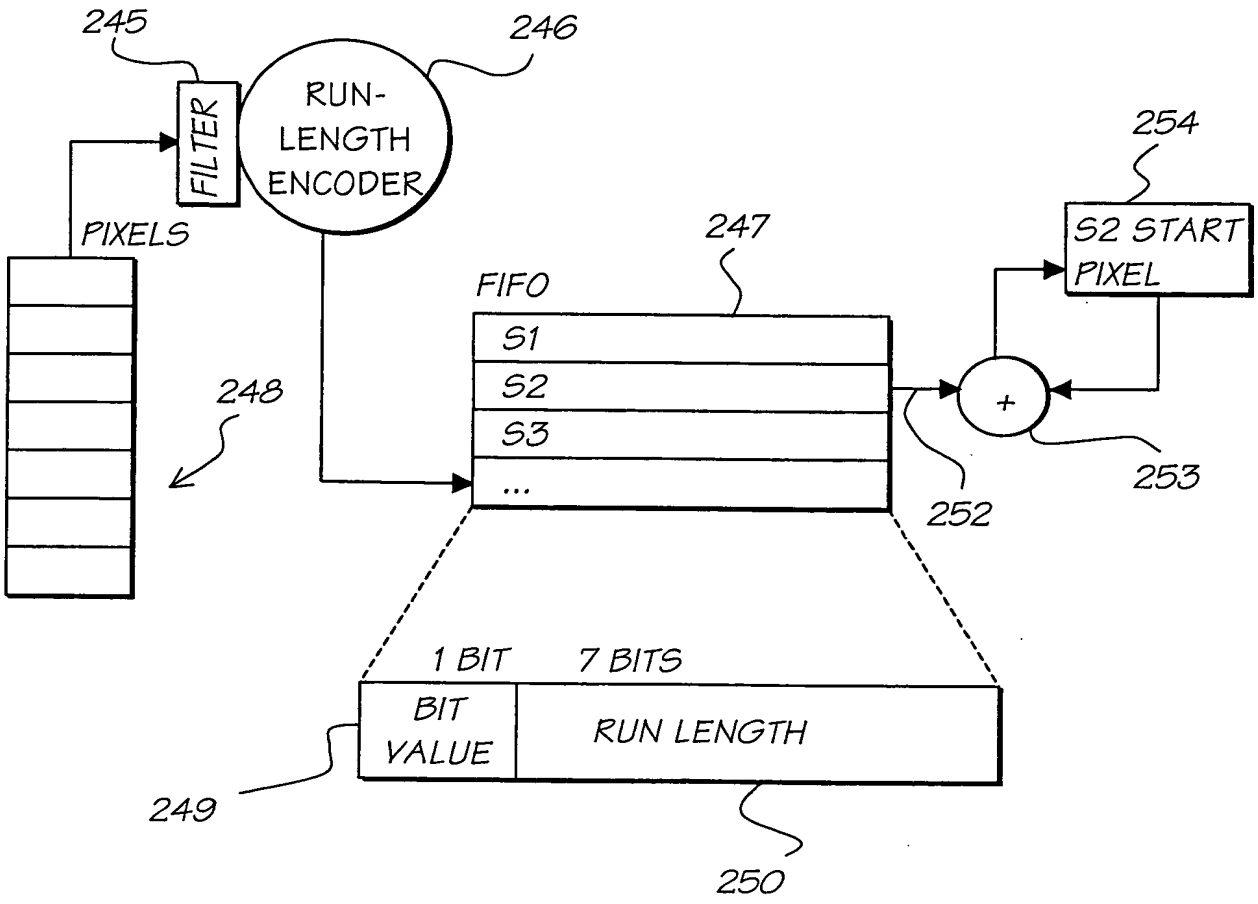

245
246
FILTER
RUN-
LENGTH
ENCODER
254
S2 START
PIXEL
PIXELS
247
FIFO
S1
S2
S3
...
248
+
253
252
1 BIT
7 BITS
BIT
VALUE
RUN LENGTH
249
250

FIG. 40

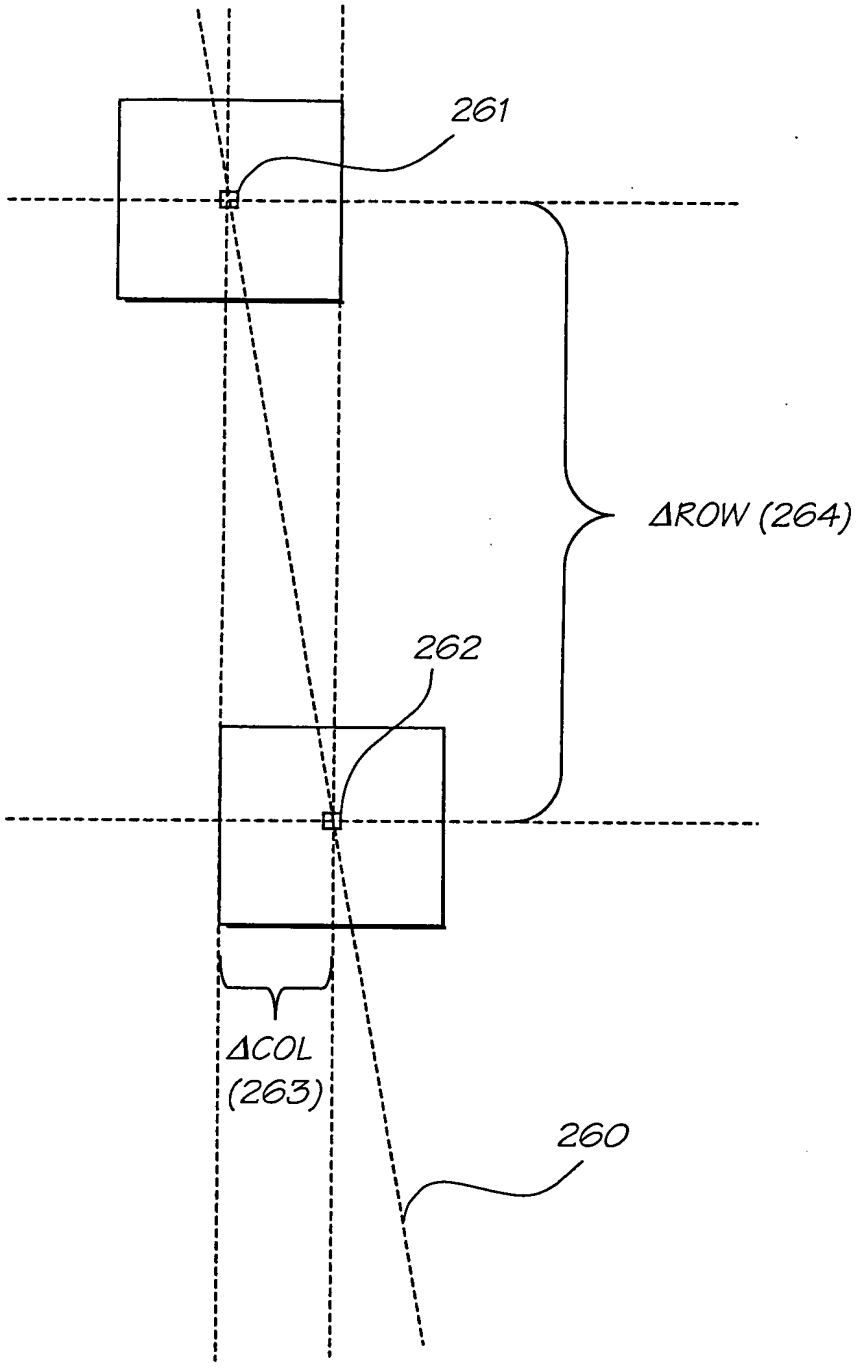


FIG. 41

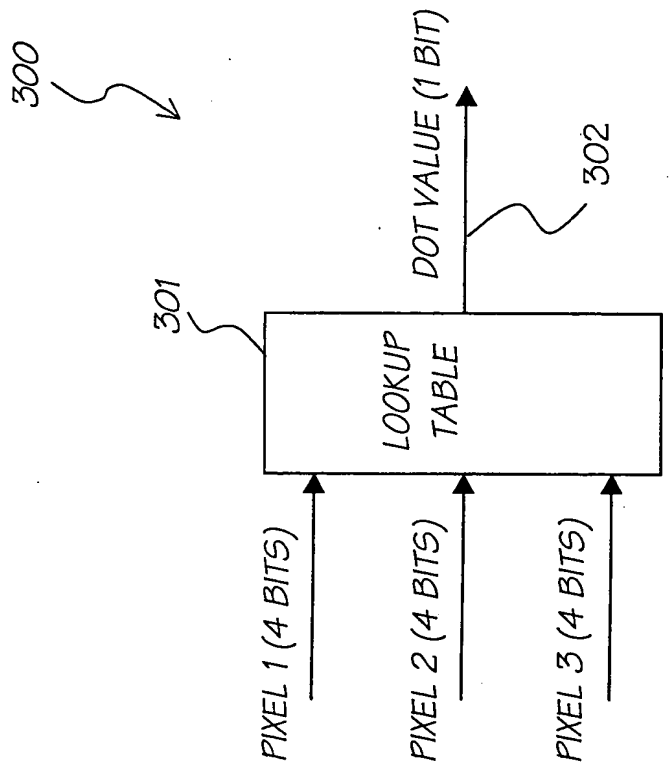

300
301
LOOKUP TABLE
DOT VALUE (1 BIT)
302
PIXEL 1 (4 BITS)
PIXEL 2 (4 BITS)
PIXEL 3 (4 BITS)

FIG. 43

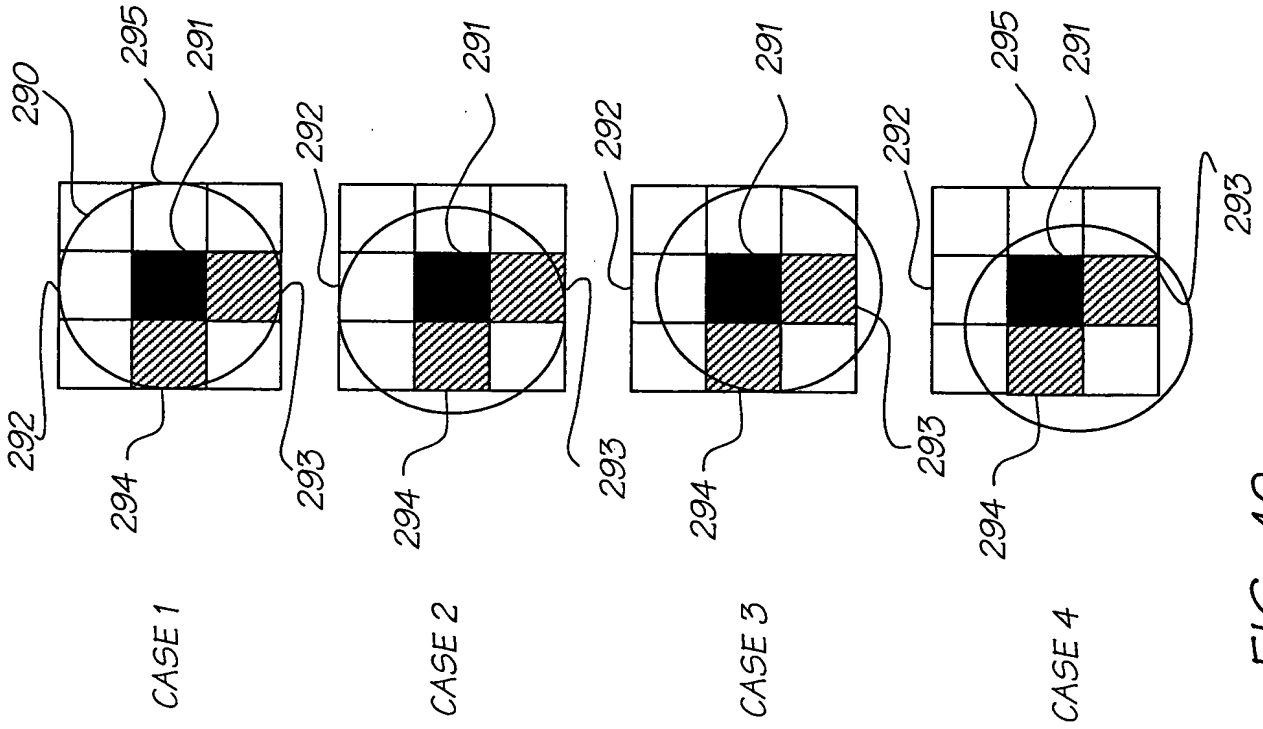

290
291
292
293
294
295
CASE 1
CASE 2
CASE 3
CASE 4

FIG. 42

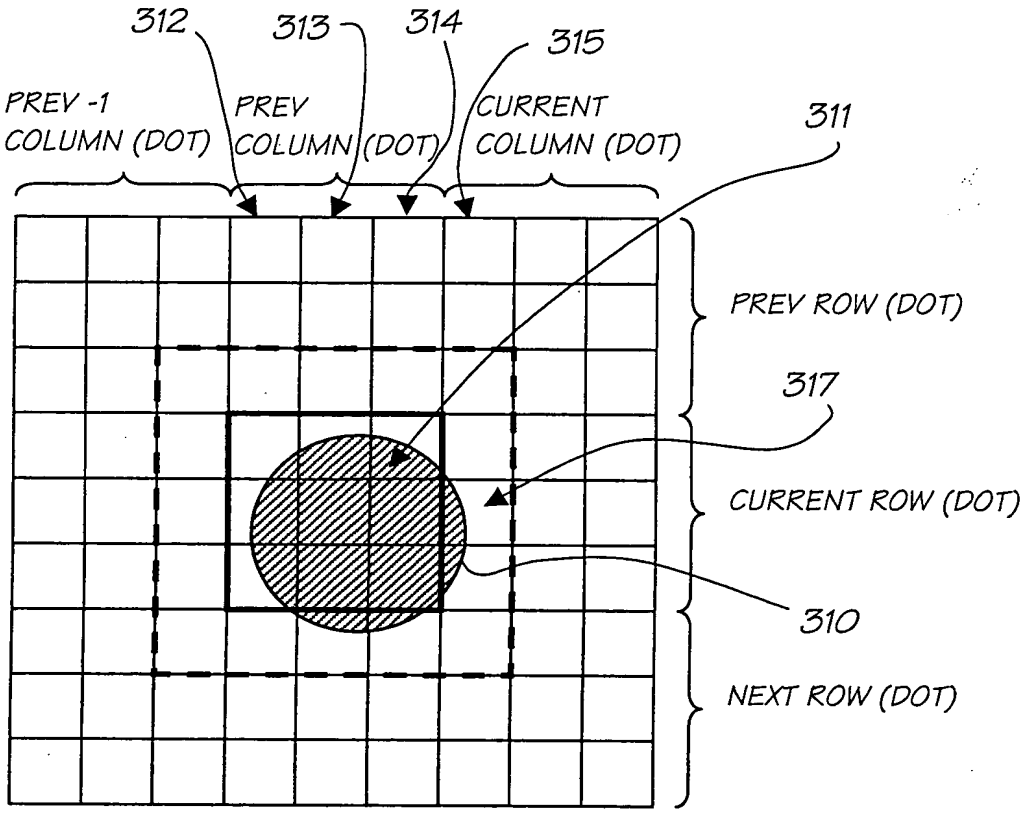

312
313
314
315
311
PREV -1 COLUMN (DOT)
PREV COLUMN (DOT)
CURRENT COLUMN (DOT)
PREV ROW (DOT)
317
CURRENT ROW (DOT)
310
NEXT ROW (DOT)

FIG. 44

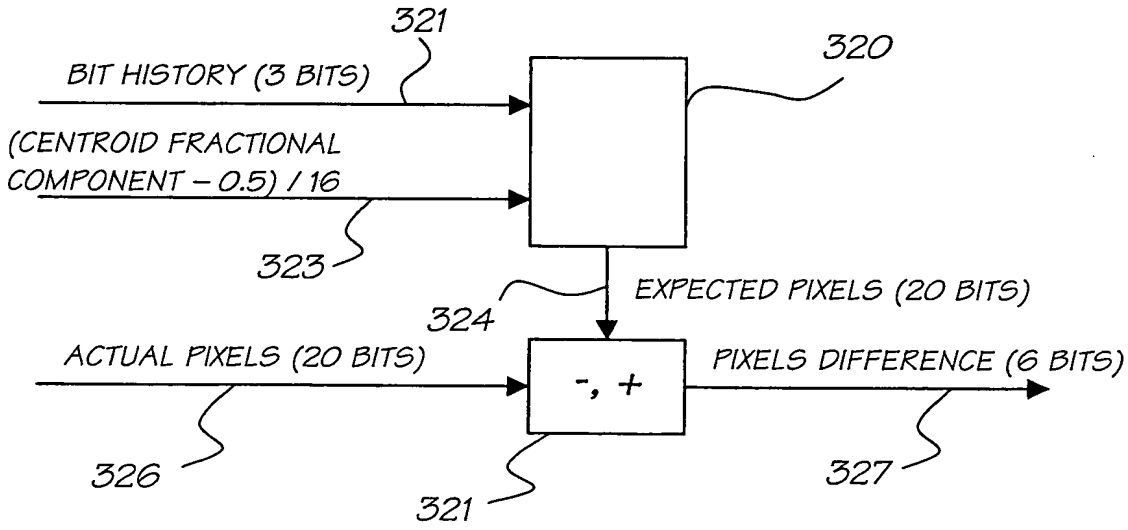

321
320
BIT HISTORY (3 BITS)
(CENTROID FRACTIONAL COMPONENT – 0.5) / 16
323
324
EXPECTED PIXELS (20 BITS)
ACTUAL PIXELS (20 BITS)
-, +
PIXELS DIFFERENCE (6 BITS)
326
321
327

FIG. 45

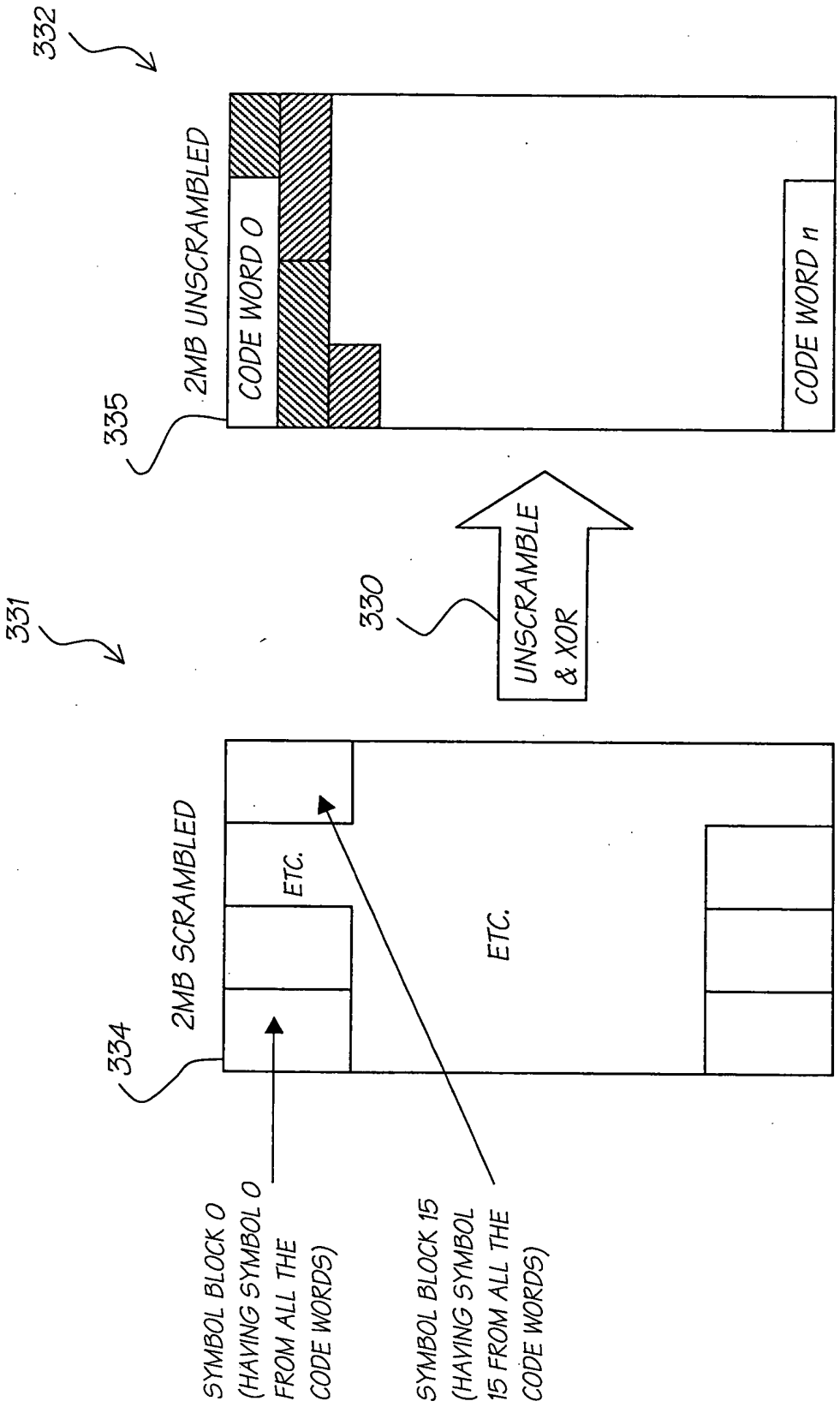
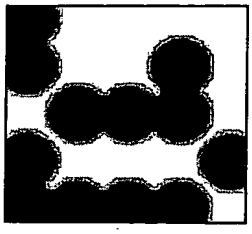
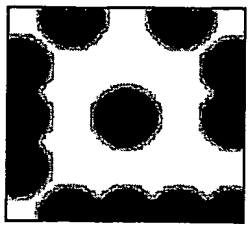

332
2MB UNSCRAMBLED
CODE WORD 0
CODE WORD n
335
330
UNSCRAMBLE & XOR
331
2MB SCRAMBLED
ETC.
ETC.
334
SYMBOL BLOCK 0 (HAVING SYMBOL 0 FROM ALL THE CODE WORDS)
SYMBOL BLOCK 15 (HAVING SYMBOL 15 FROM ALL THE CODE WORDS)

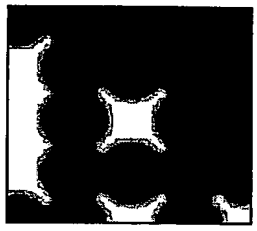
FIG. 46



BLACK AND WHITE DOTS



BLACK DOT SURROUNDED BY WHITE



WHITE DOT SURROUNDED BY BLACK

FIG. 47

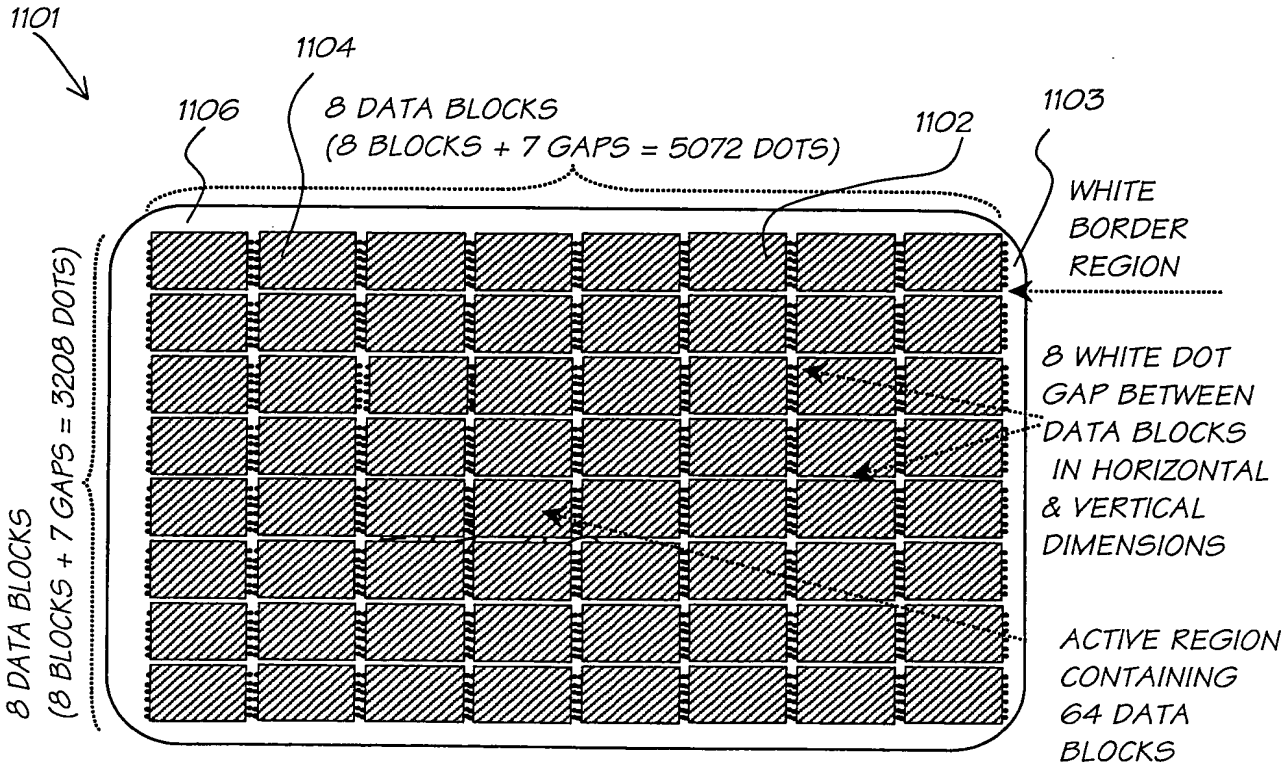


FIG. 48

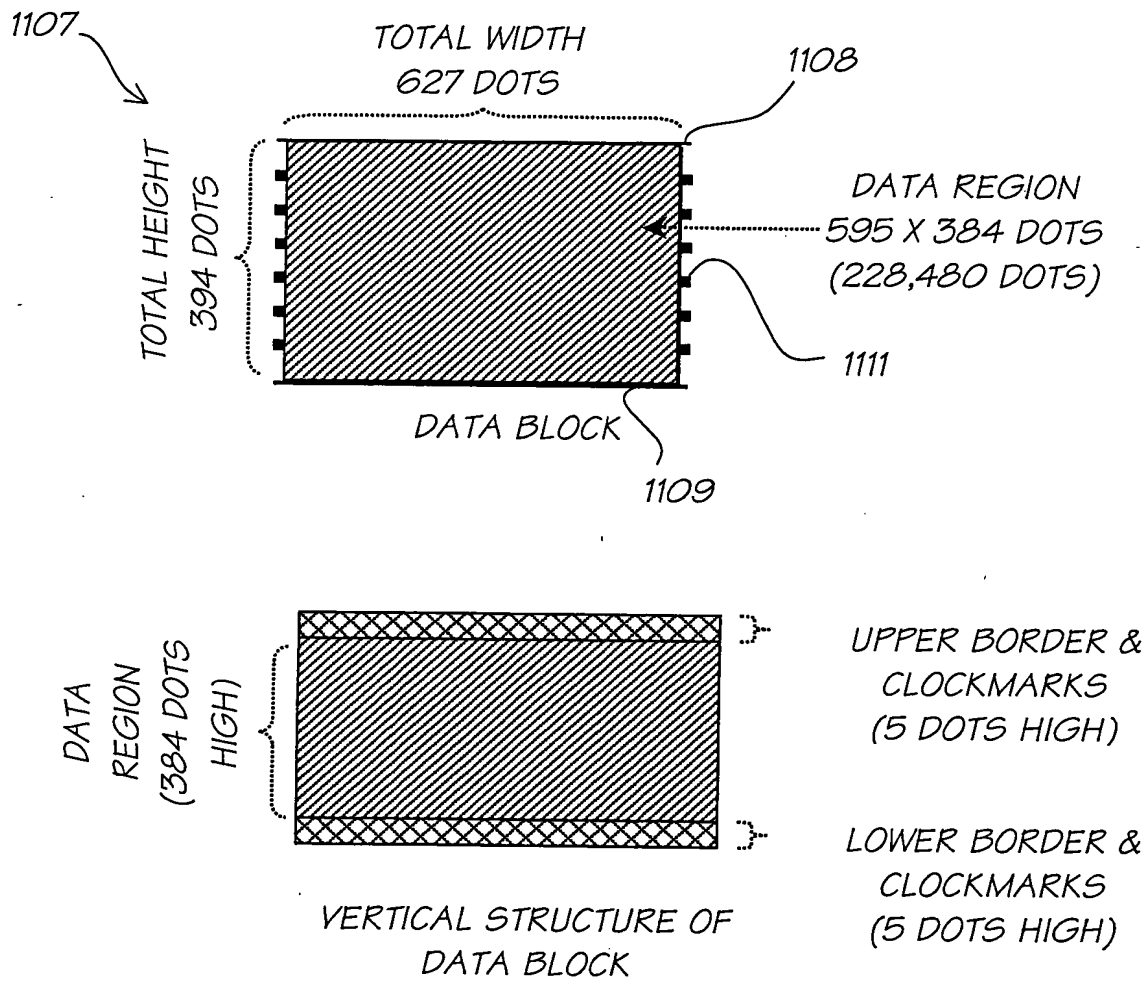

1107
TOTAL WIDTH
627 DOTS
1108
TOTAL HEIGHT
394 DOTS
DATA REGION
595 X 384 DOTS
(228,480 DOTS)
1111
DATA BLOCK
1109
DATA
REGION
(384 DOTS
HIGH)
UPPER BORDER &
CLOCKMARKS
(5 DOTS HIGH)
LOWER BORDER &
CLOCKMARKS
(5 DOTS HIGH)
VERTICAL STRUCTURE OF
DATA BLOCK
LEFT TARGETS
(15 DOTS WIDE)
DATA REGION
(595 DOTS WIDE)
RIGHT TARGETS
(15 DOTS WIDE)
WHITE DOT
ORIENTATION
COLUMN
(1 DOT WIDE)
BLACK DOT
ORIENTATION
COLUMN
(1 DOT WIDE)
HORIZONTAL STRUCTURE
OF DATA BLOCK

FIG. 49

FIG. 51

FIG. 52

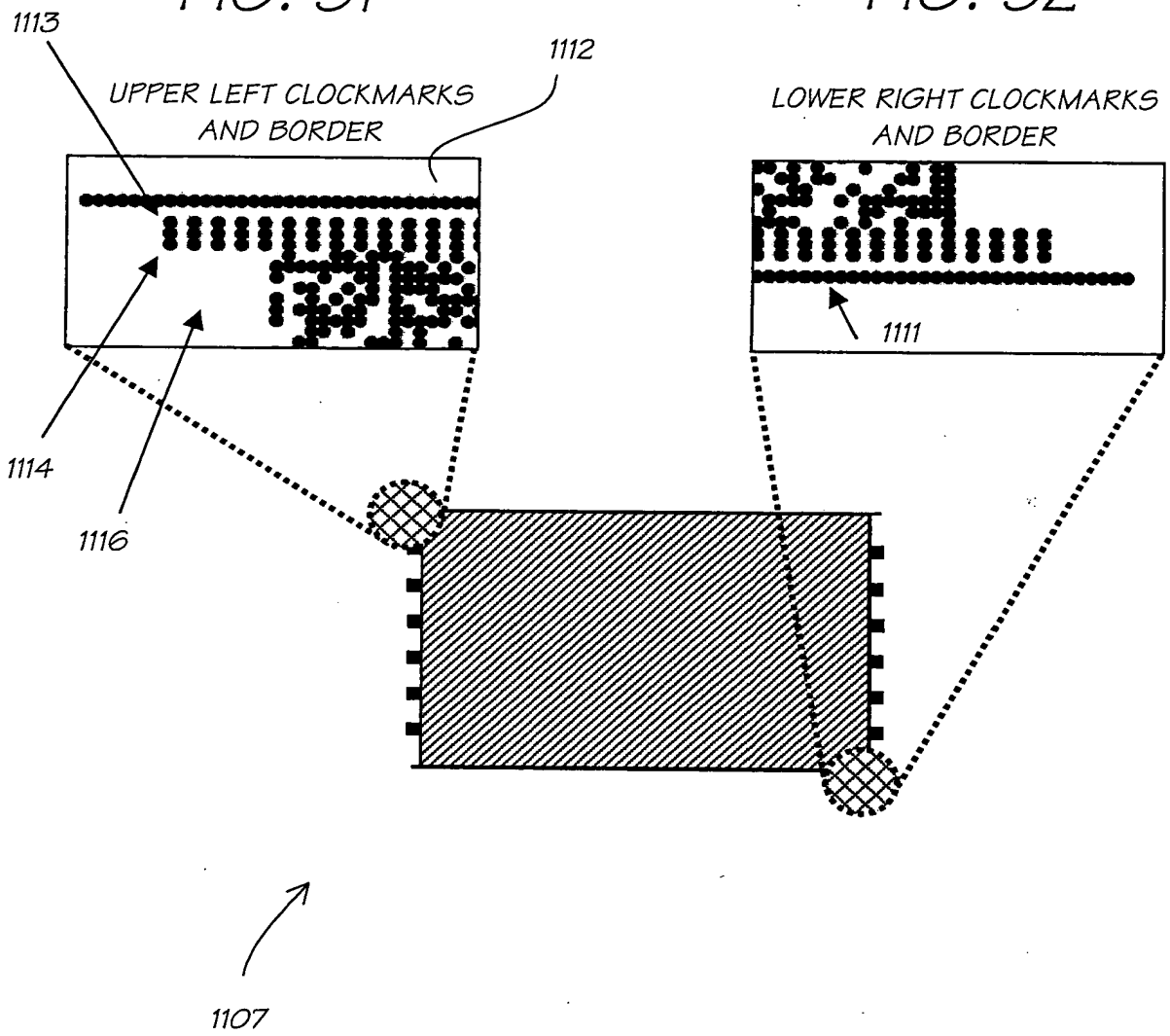


FIG. 50

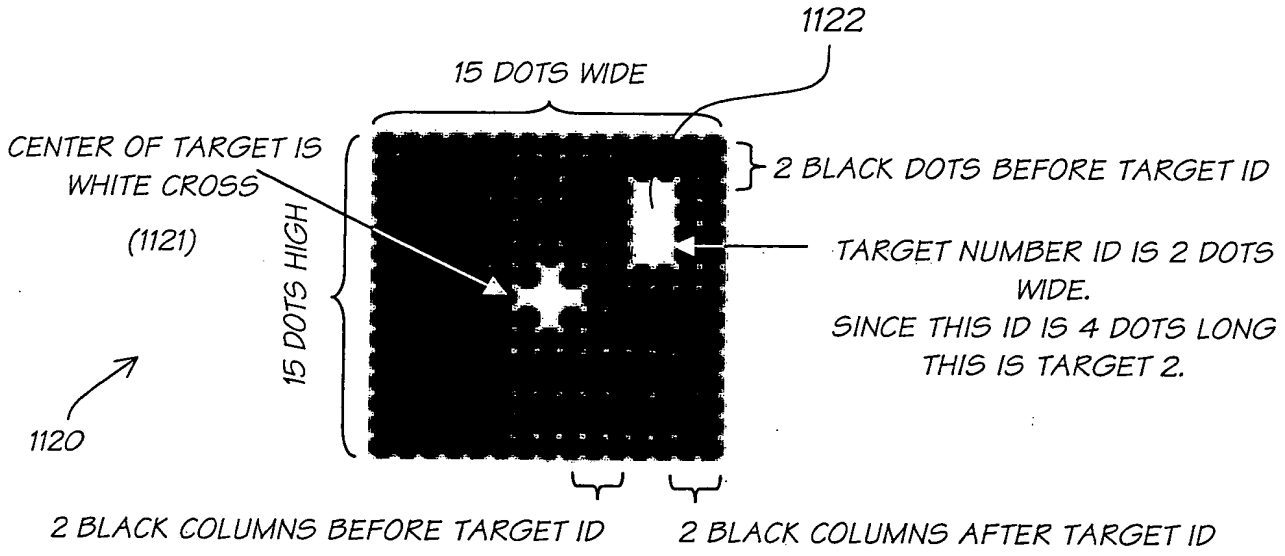

1122
15 DOTS WIDE
CENTER OF TARGET IS WHITE CROSS
(1121)
2 BLACK DOTS BEFORE TARGET ID
TARGET NUMBER ID IS 2 DOTS WIDE.
SINCE THIS ID IS 4 DOTS LONG THIS IS TARGET 2.
15 DOTS HIGH
1120
2 BLACK COLUMNS BEFORE TARGET ID
2 BLACK COLUMNS AFTER TARGET ID

FIG. 53

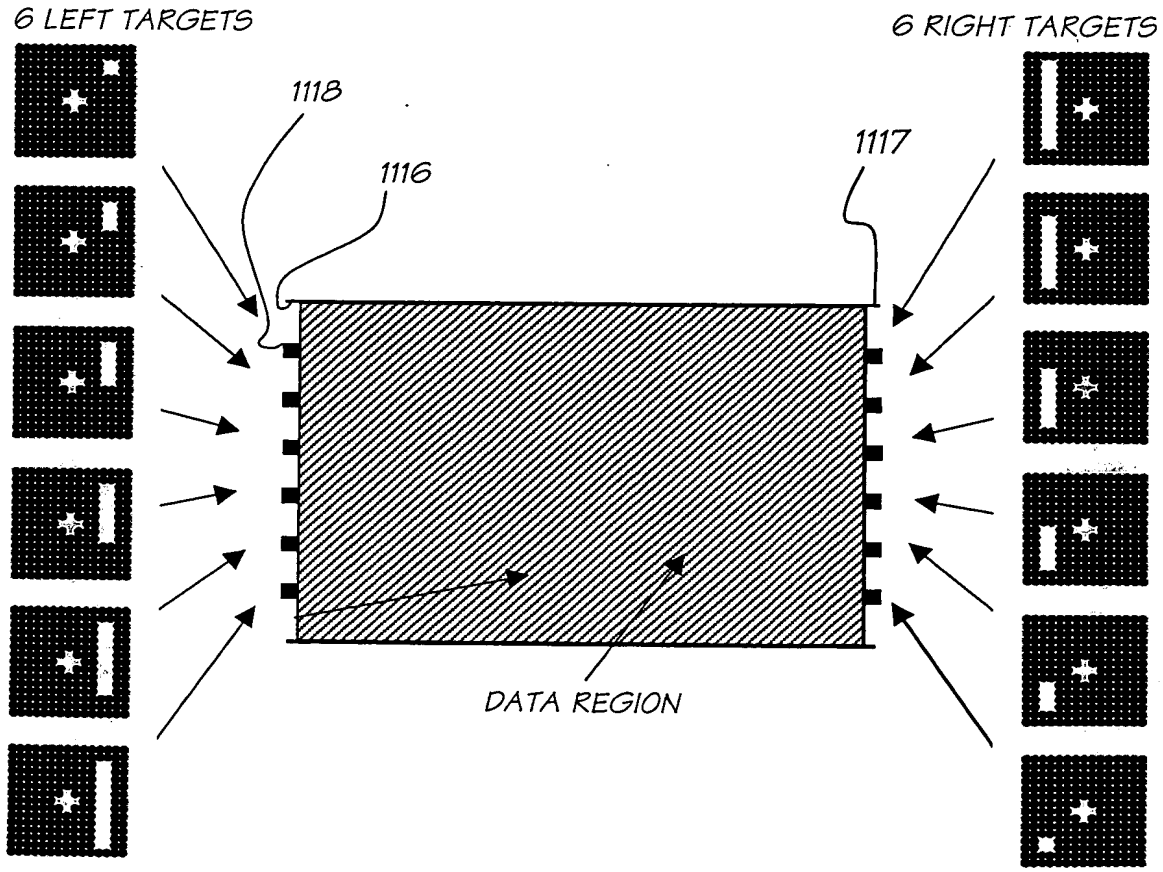

6 LEFT TARGETS
6 RIGHT TARGETS
1118
1116
1117
DATA REGION

FIG. 54

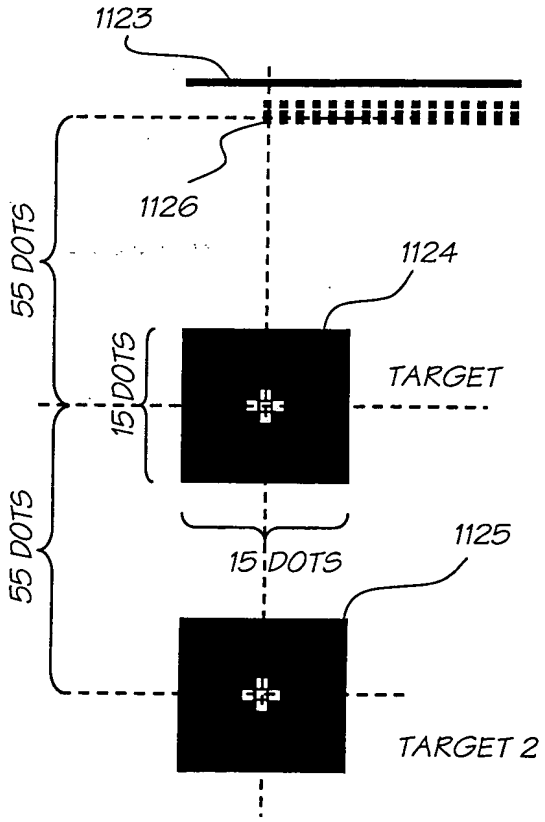

1123
1126
55 DOTS
1124
15 DOTS
TARGET
55 DOTS
15 DOTS
1125
TARGET 2

FIG. 55

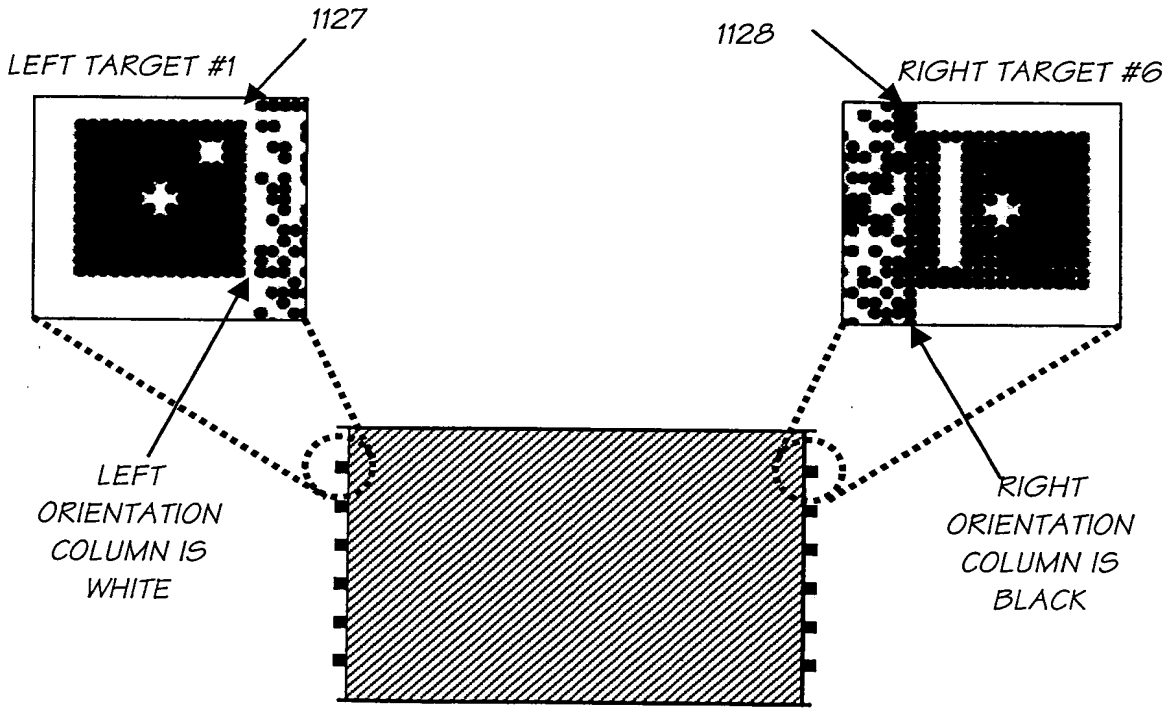

1127
LEFT TARGET #1
1128
RIGHT TARGET #6
LEFT ORIENTATION COLUMN IS WHITE
RIGHT ORIENTATION COLUMN IS BLACK

FIG. 56

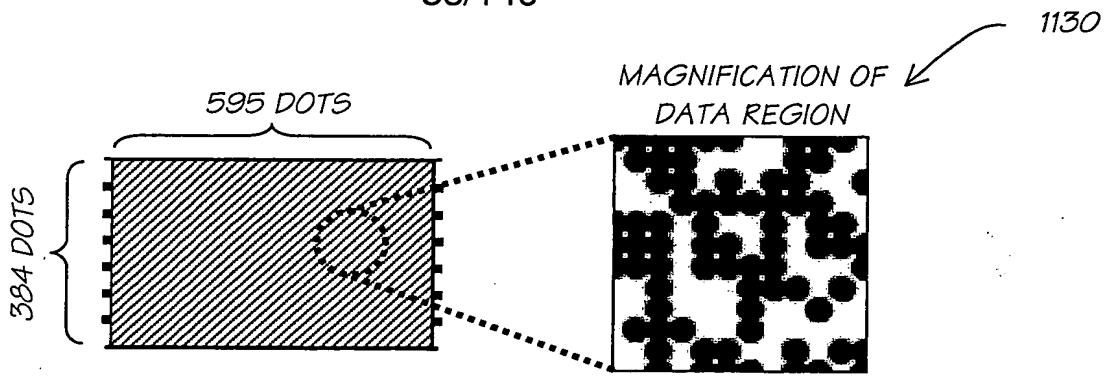


FIG. 57

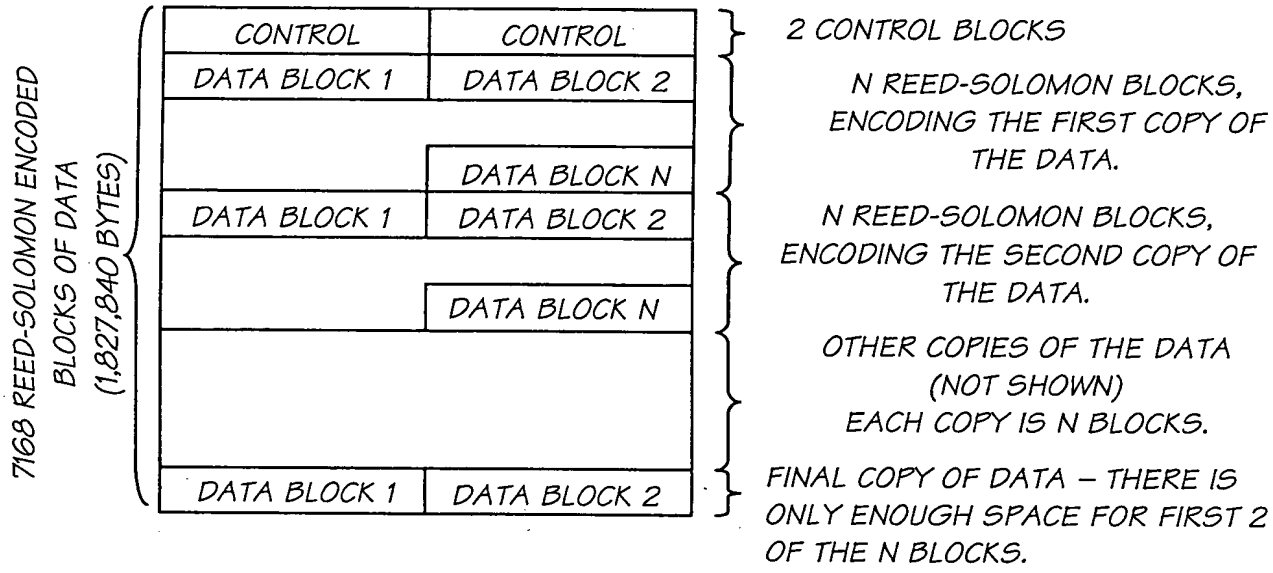


FIG. 58

```
00: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
0C: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
18: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
24: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
30: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
3C: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
48: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
54: 4F 00 3D 4F 00 3D 4F 00 3D 4F 00 3D
60: 00 00 00 00 00 00 00 00 00 00 00 00
6C: 00 00 00 00 00 00 00 00 00 00 00 00
78: 00 00 00 00 00 00 00 00 00 00 00 00
```

32 COPIES OF THE 3 BYTE CONTROL INFORMATION

RESERVED BYTES ARE 0

FIG. 59

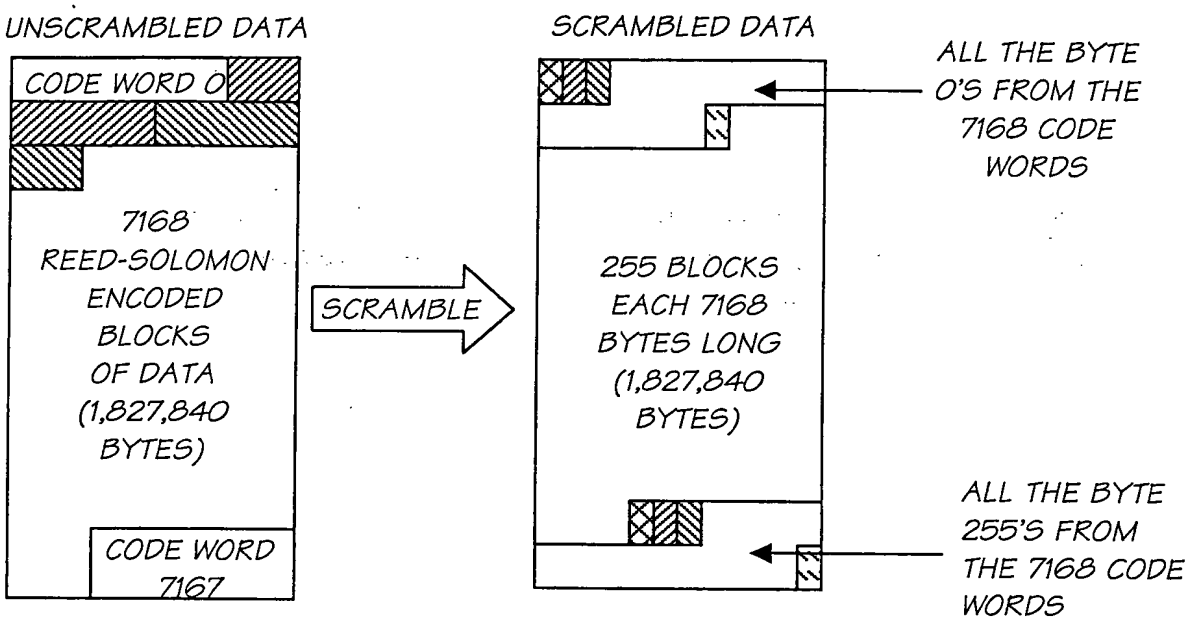


FIG. 60

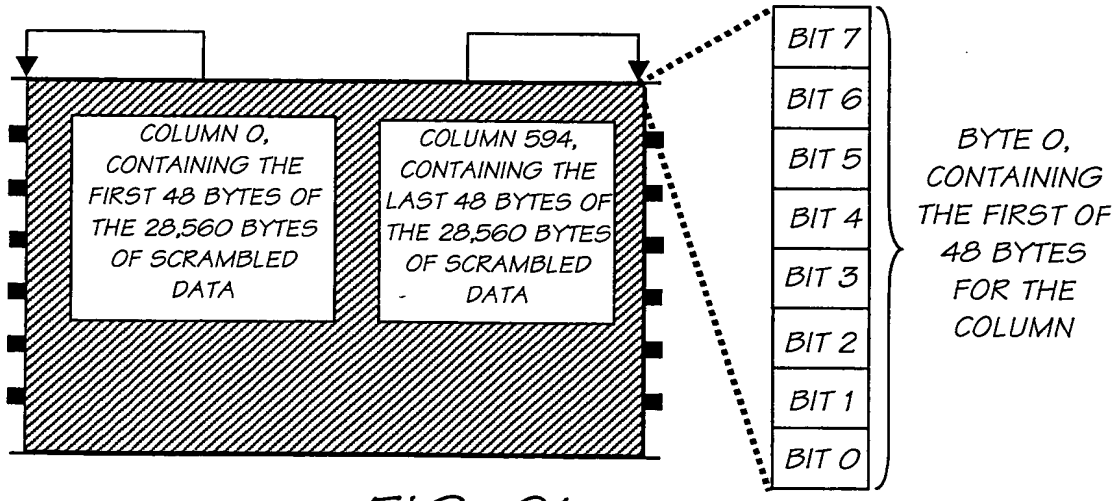


FIG. 61

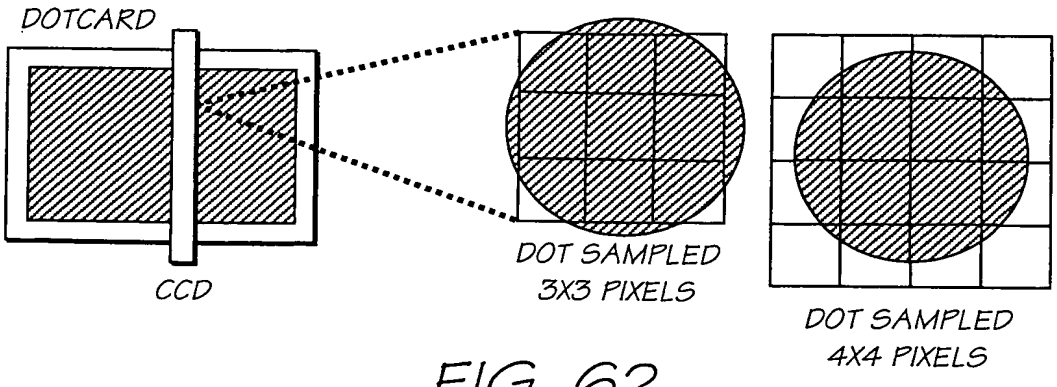


FIG. 62

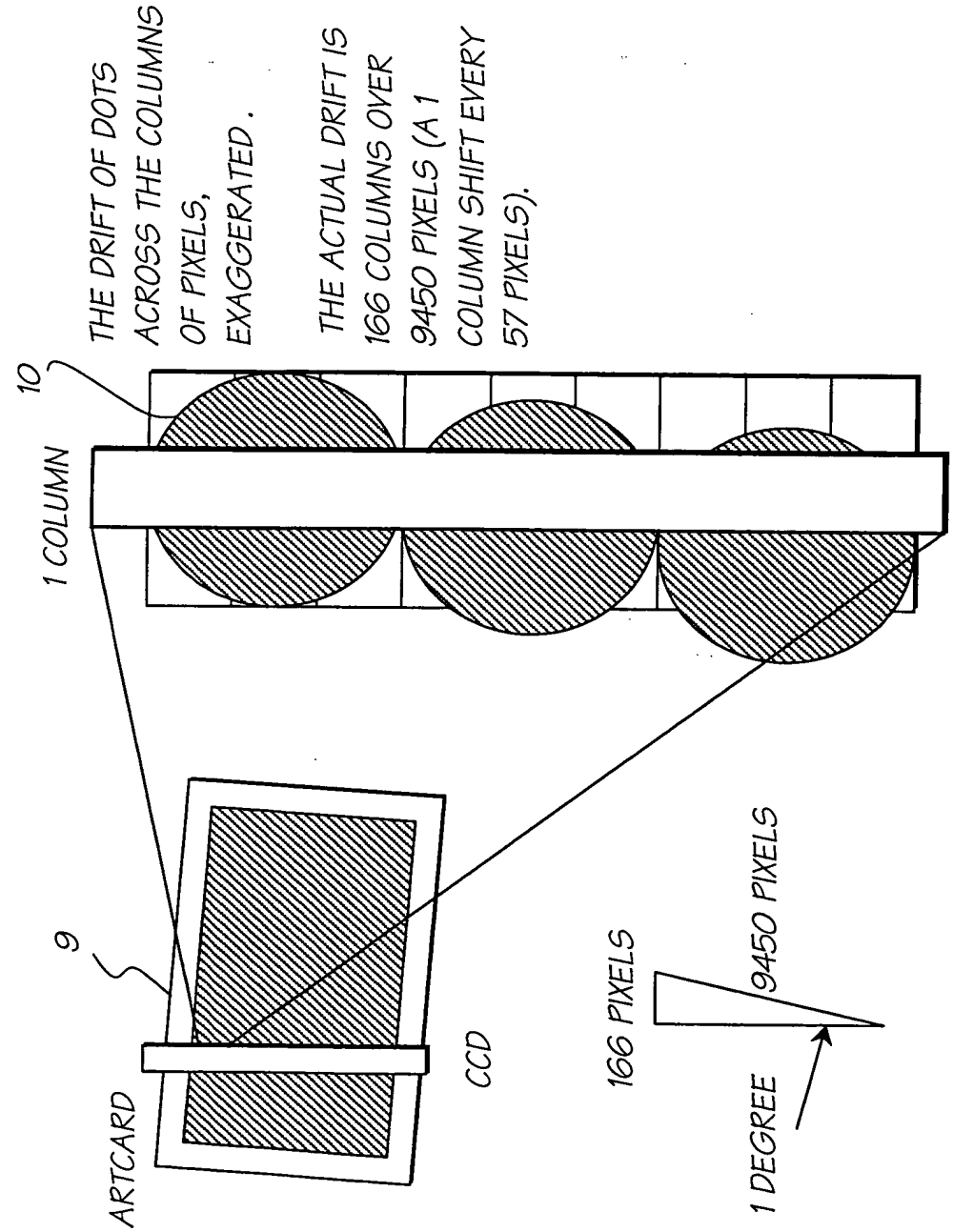

THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).
10
1 COLUMN
9
ARTCARD
CCD
166 PIXELS
9450 PIXELS
1 DEGREE

FIG. 63

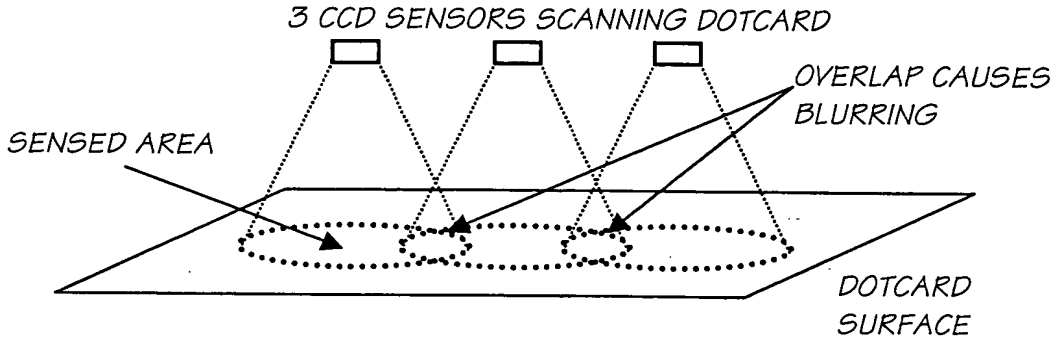


FIG. 64

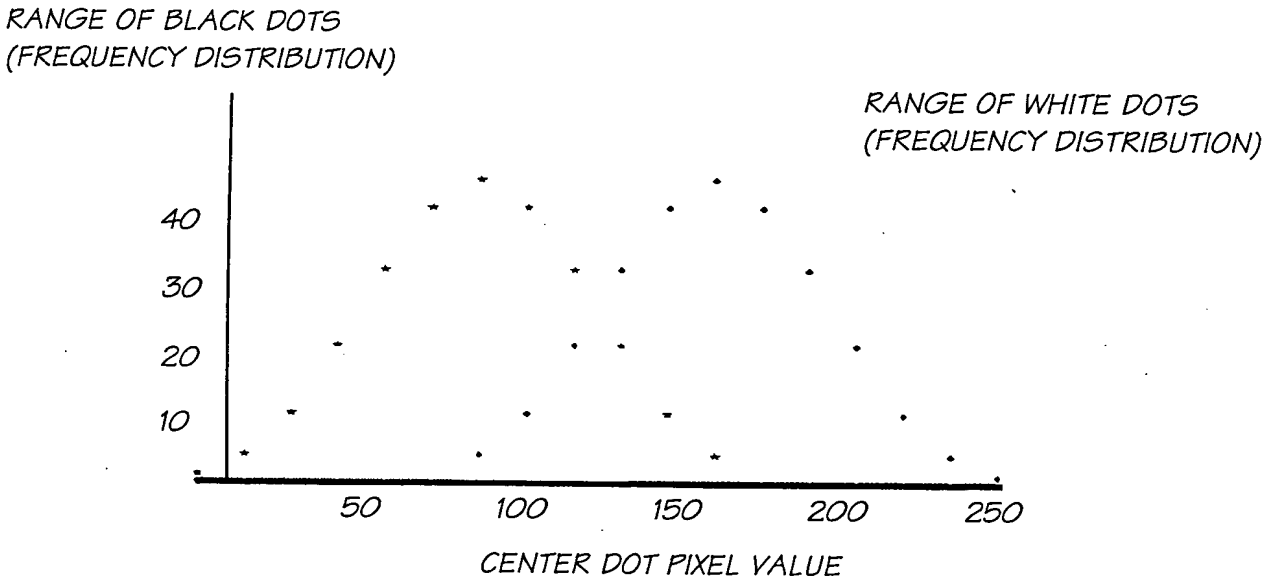


FIG. 65

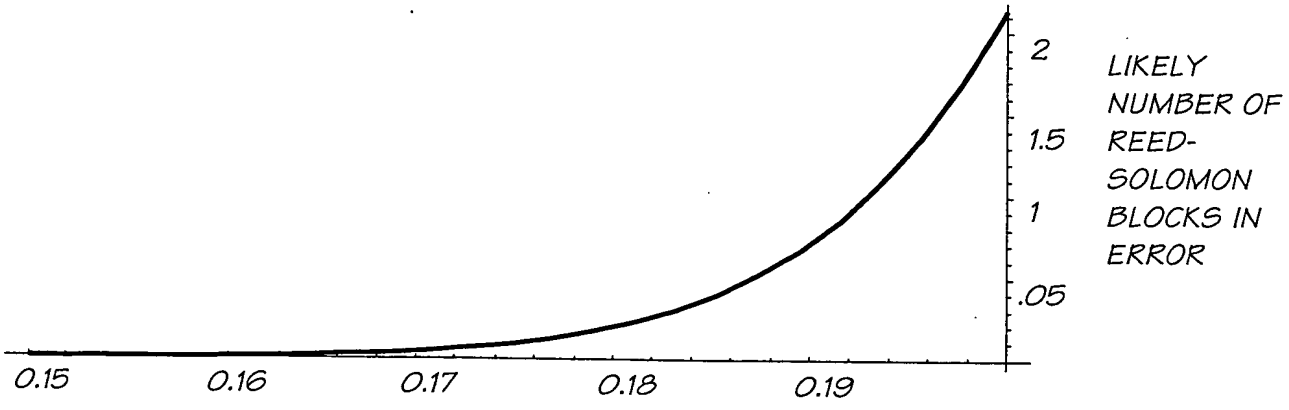
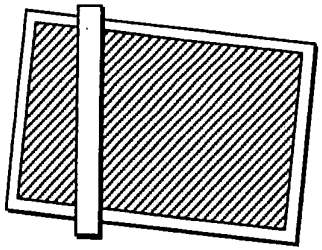


FIG. 66



READ LINEAR CCD

DOTCARD PIXEL DATA

EXTRACT BIT IMAGE

ENCODED, SCRAMBLED, POTENTIALLY ROTATED 180° BITMAPPED DATA

ROTATE AND UNSCRAMBLE

ENCODED, UNSCRAMBLED DATA

REED-SOLOMON DECODE

RAW DATA

APPROXIMATE DATA SIZES FOR 1600 DPI DOTCARD

86MM + 1MM IN HORIZONTAL DIMENSION FOR 1° ROTATION = 87MM
87MM = 16,252 SCANLINES
16,440 SCANLINES @ 11,000 PIXELS PER SCANLINE = 180,840,000 PIXELS
180,840,000 PIXELS @ 1 BYTE PER PIXEL = 180,840,000 BYTES = 172.5 MB

64 DATA BLOCKS, EACH CONTAINING 597 COLUMNS (595 DATA REGION COLUMNS AND 2 ORIENTATION COLUMNS), @ 48 BYTES PER COLUMN = 28,656 BYTES PER DATA BLOCK FOR A TOTAL OF 1,833,984 BYTES.

64 DATA BLOCKS, EACH CONTAINING 112 ENCODED REED SOLOMON BLOCKS, @ 255 BYTES PER REED SOLOMON BLOCK FOR A TOTAL OF 1,827,840 BYTES.

DECODED DATA, WITH A MAXIMUM SIZE OF 910,082 BYTES.
(64 X 112 X 127 – (2 CONTROL BLOCKS @ 127 BYTES))

FIG. 67

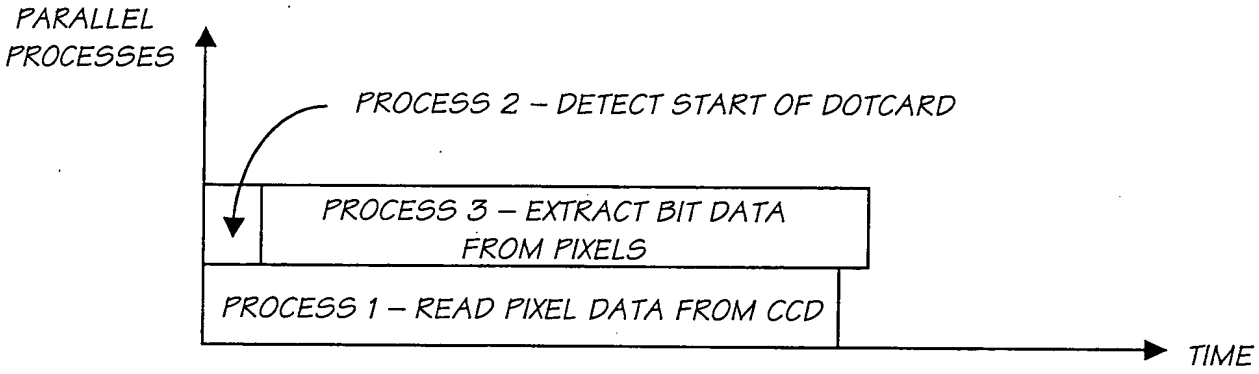

PARALLEL PROCESSES
PROCESS 2 – DETECT START OF DOTCARD
PROCESS 3 – EXTRACT BIT DATA FROM PIXELS
PROCESS 1 – READ PIXEL DATA FROM CCD
TIME

FIG. 68

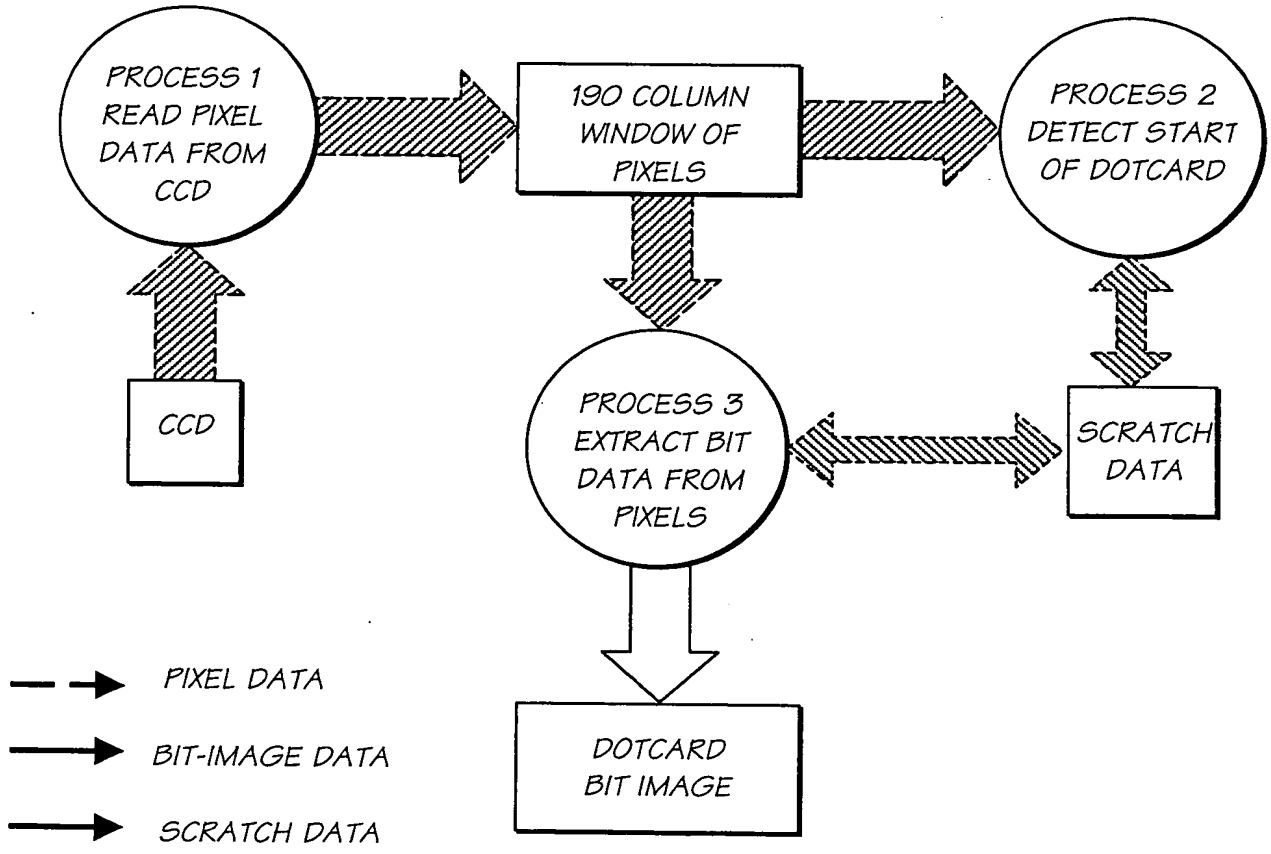

PROCESS 1 READ PIXEL DATA FROM CCD
190 COLUMN WINDOW OF PIXELS
PROCESS 2 DETECT START OF DOTCARD
CCD
PROCESS 3 EXTRACT BIT DATA FROM PIXELS
SCRATCH DATA
DOTCARD BIT IMAGE
PIXEL DATA
BIT-IMAGE DATA
SCRATCH DATA

FIG. 69

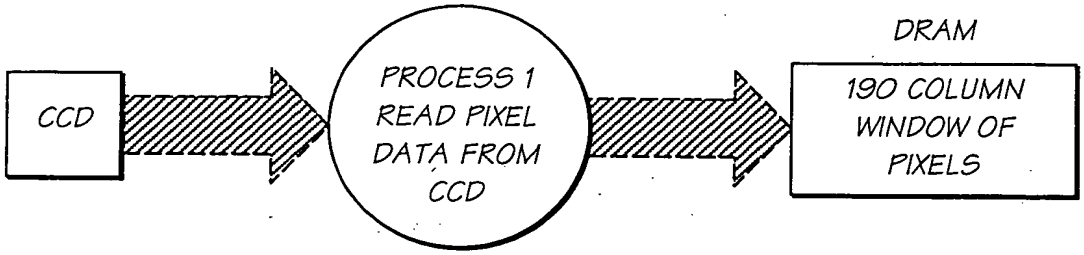


FIG. 70

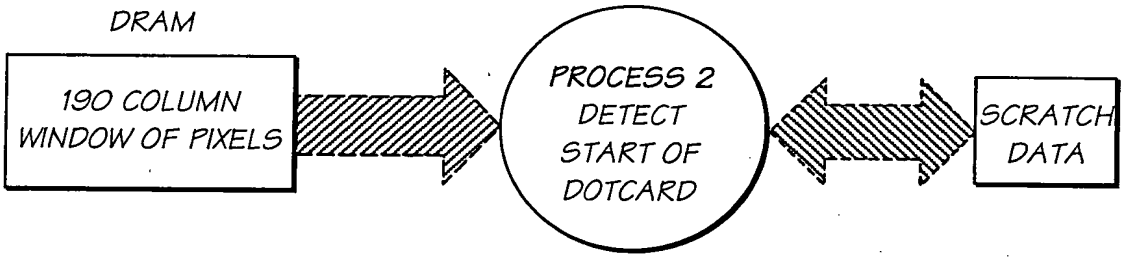


FIG. 71

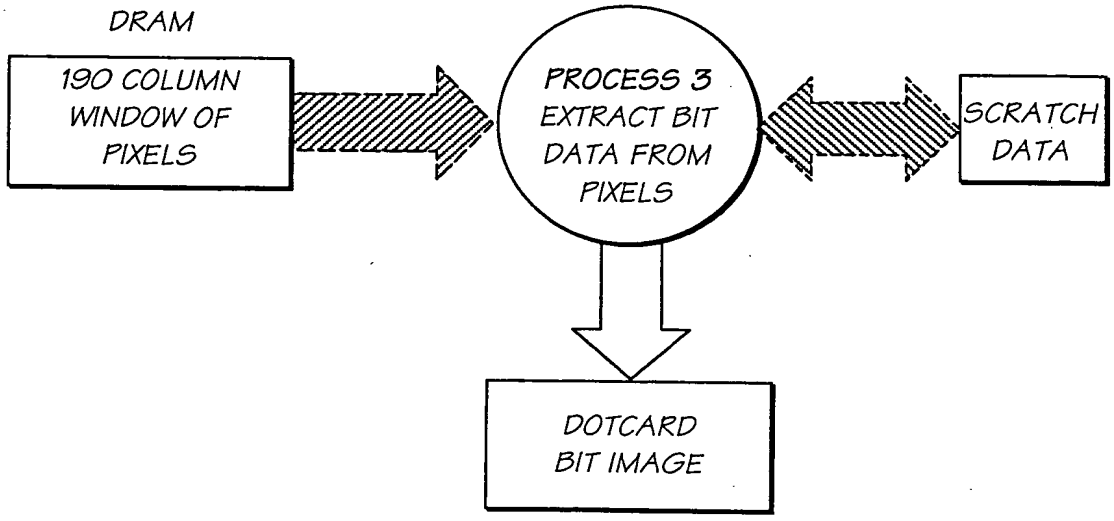


FIG. 72

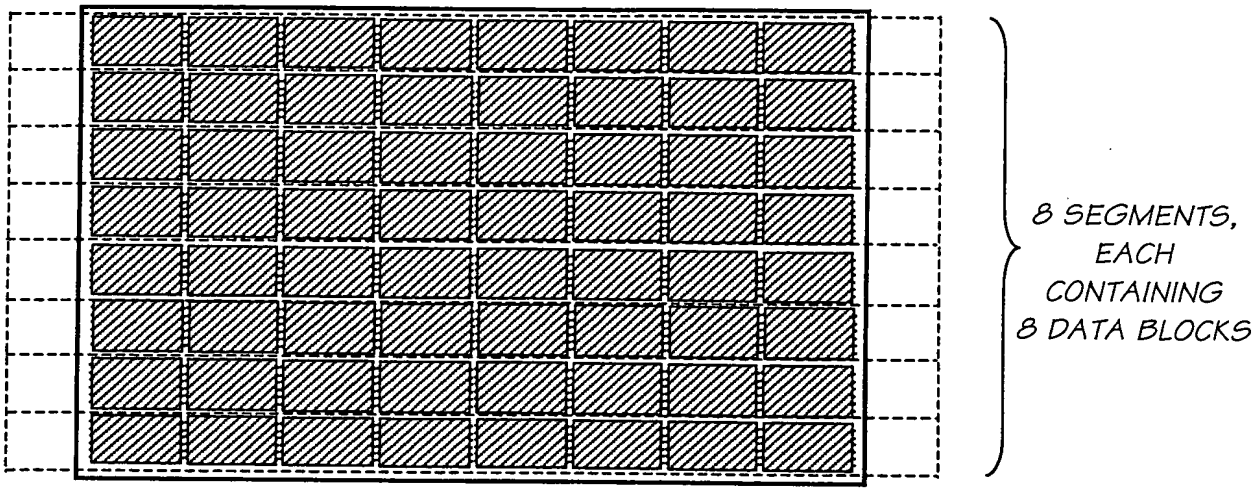

8 SEGMENTS,
EACH
CONTAINING
8 DATA BLOCKS

FIG. 73

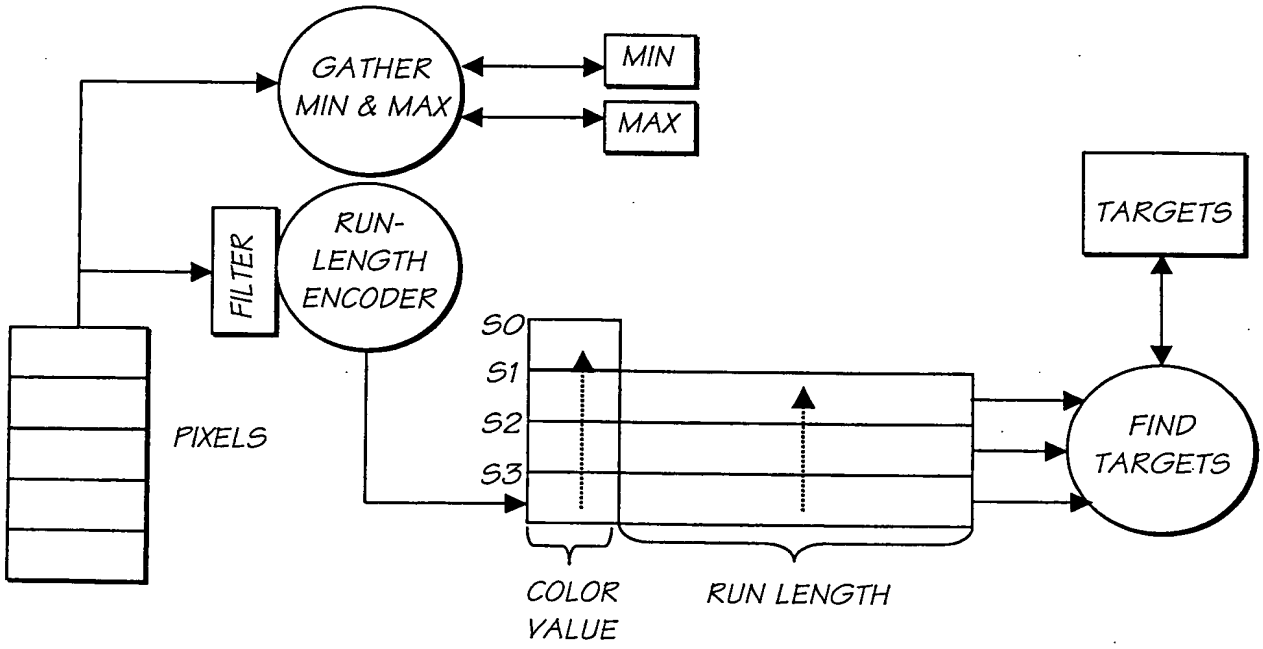

GATHER
MIN & MAX
MIN
MAX
FILTER
RUN-
LENGTH
ENCODER
TARGETS
PIXELS
S0
S1
S2
S3
FIND
TARGETS
COLOR
VALUE
RUN LENGTH

FIG. 74

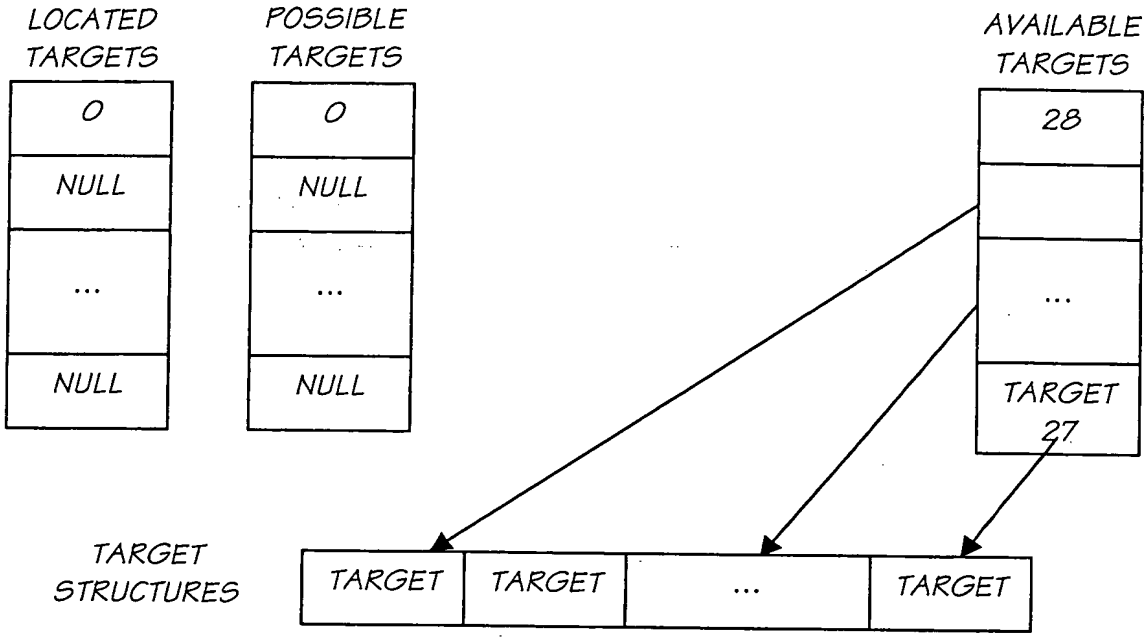
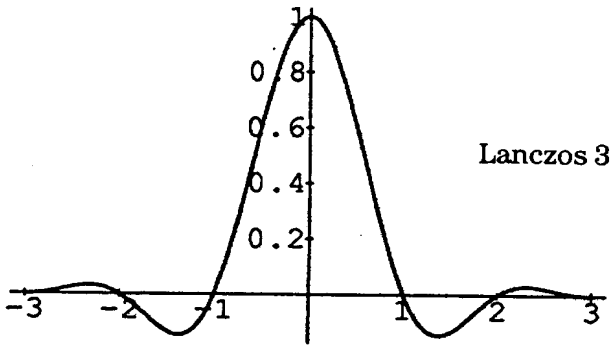


FIG. 75



$$\text{Lanczos } 3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x} \dfrac{\sin\left(\pi \dfrac{x}{3}\right)}{\pi \dfrac{x}{3}}, & |x| < 3 \\ 0, & |x| \geq 3 \end{cases}$$

FIG. 76

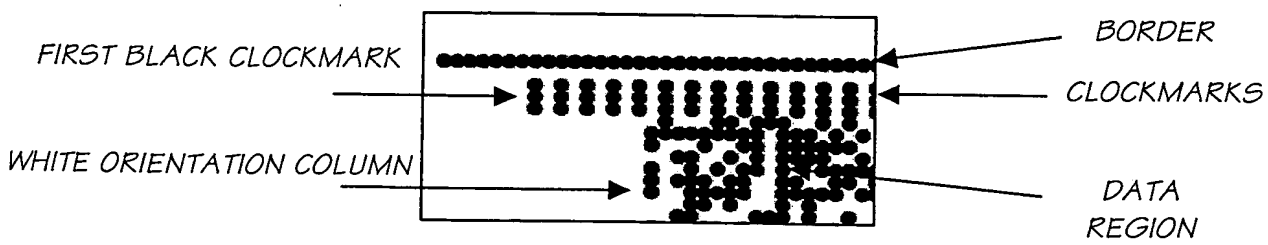


FIG. 77

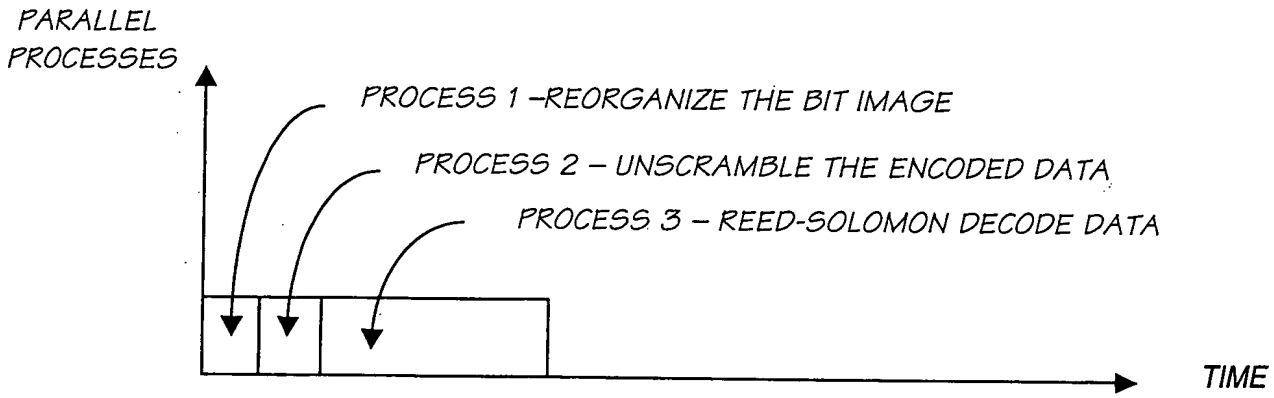

PARALLEL PROCESSES
PROCESS 1 –REORGANIZE THE BIT IMAGE
PROCESS 2 – UNSCRAMBLE THE ENCODED DATA
PROCESS 3 – REED-SOLOMON DECODE DATA
TIME

FIG. 78

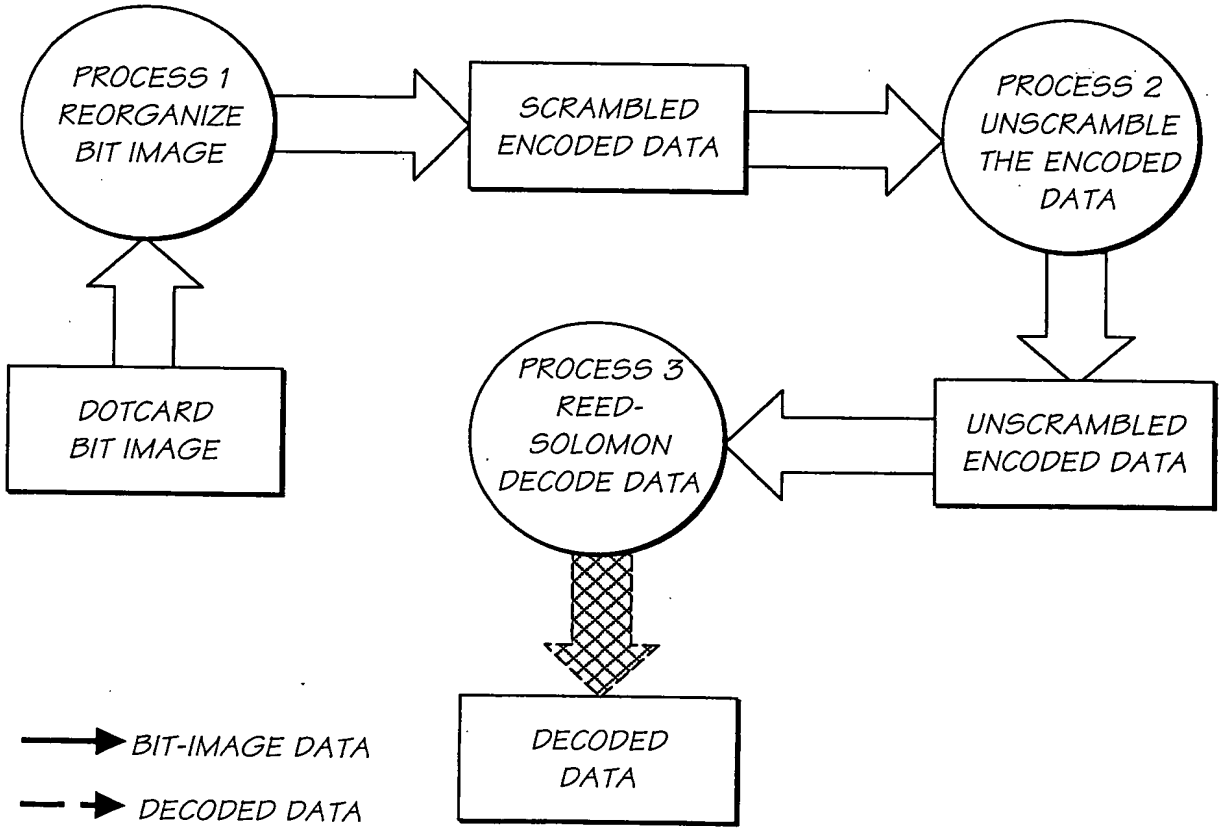

PROCESS 1 REORGANIZE BIT IMAGE
SCRAMBLED ENCODED DATA
PROCESS 2 UNSCRAMBLE THE ENCODED DATA
DOTCARD BIT IMAGE
PROCESS 3 REED-SOLOMON DECODE DATA
UNSCRAMBLED ENCODED DATA
DECODED DATA
BIT-IMAGE DATA
DECODED DATA

FIG. 79

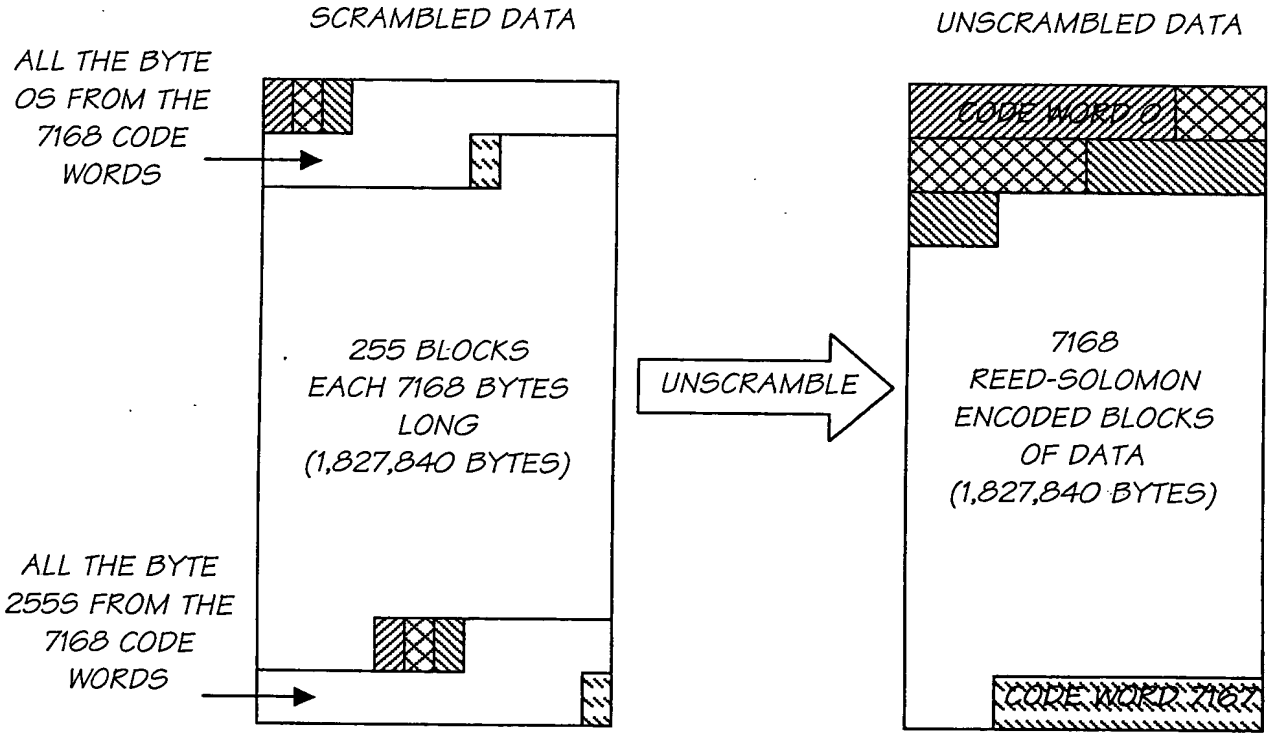

SCRAMBLED DATA
UNSCRAMBLED DATA
ALL THE BYTE OS FROM THE 7168 CODE WORDS
255 BLOCKS EACH 7168 BYTES LONG (1,827,840 BYTES)
UNSCRAMBLE
7168 REED-SOLOMON ENCODED BLOCKS OF DATA (1,827,840 BYTES)
CODE WORD 0
CODE WORD 7167
ALL THE BYTE 255S FROM THE 7168 CODE WORDS

FIG. 80

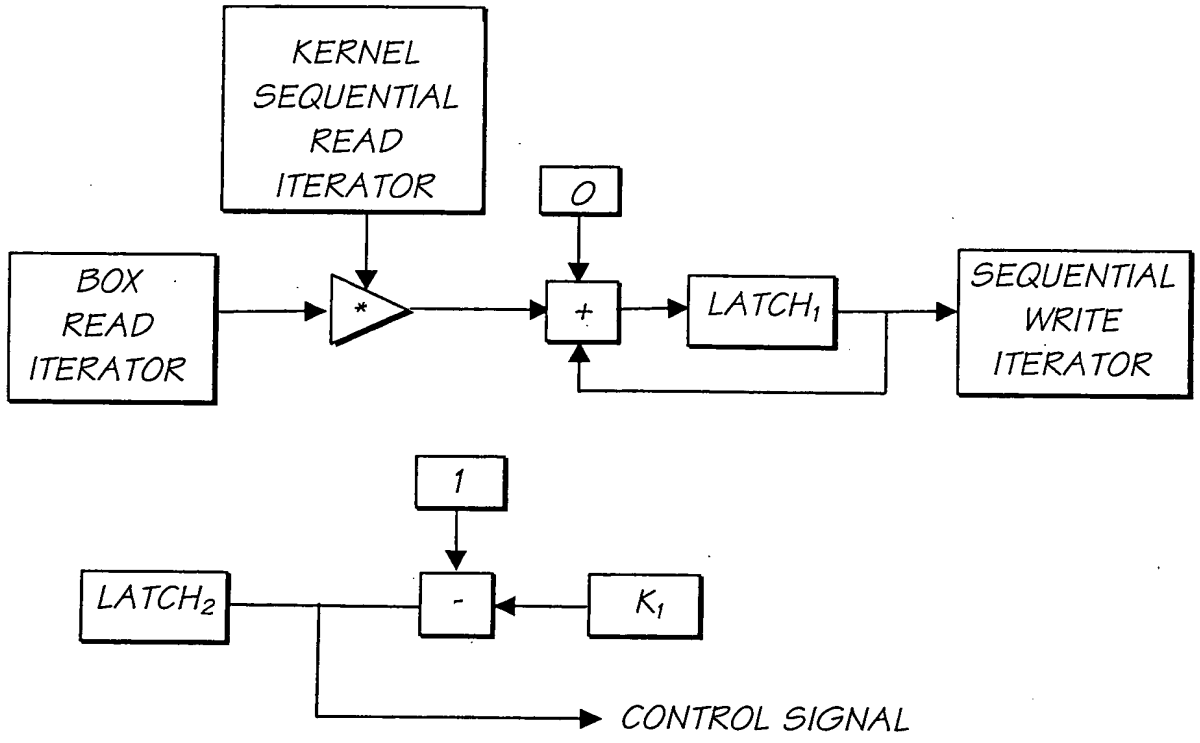

KERNEL SEQUENTIAL READ ITERATOR
0
BOX READ ITERATOR
*
+
LATCH1
SEQUENTIAL WRITE ITERATOR
1
LATCH2
-
K1
CONTROL SIGNAL

FIG. 81

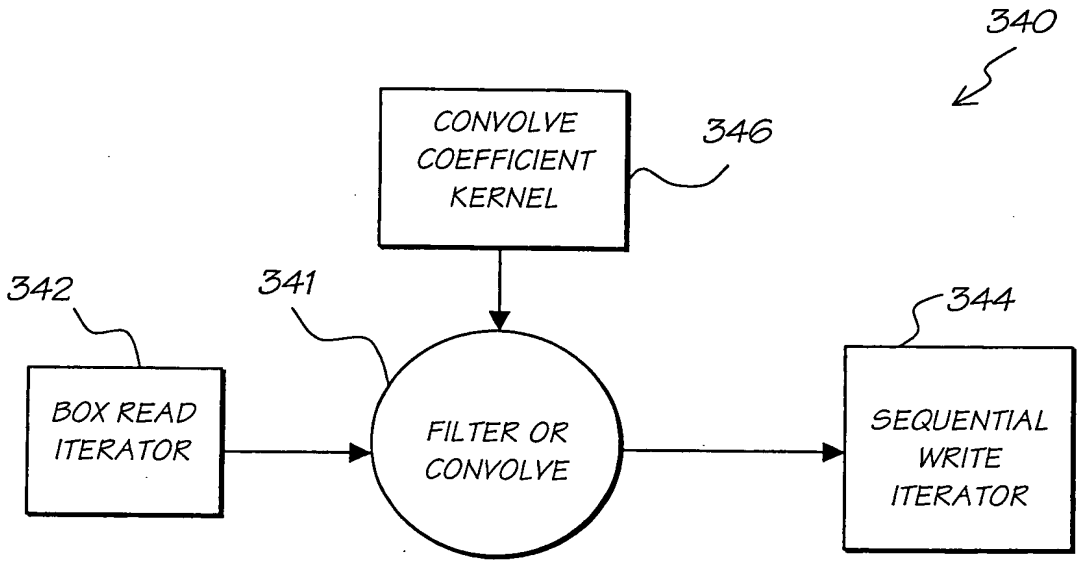

340
CONVOLVE COEFFICIENT KERNEL
346
342
341
344
BOX READ ITERATOR
FILTER OR CONVOLVE
SEQUENTIAL WRITE ITERATOR

FIG. 82

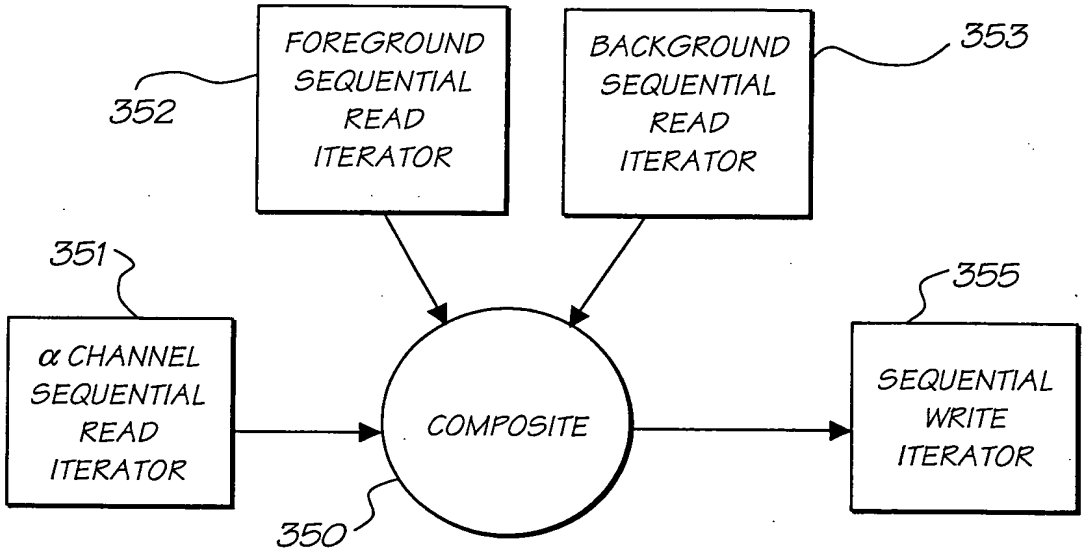

FOREGROUND SEQUENTIAL READ ITERATOR
352
BACKGROUND SEQUENTIAL READ ITERATOR
353
351
α CHANNEL SEQUENTIAL READ ITERATOR
COMPOSITE
355
SEQUENTIAL WRITE ITERATOR
350

FIG. 83

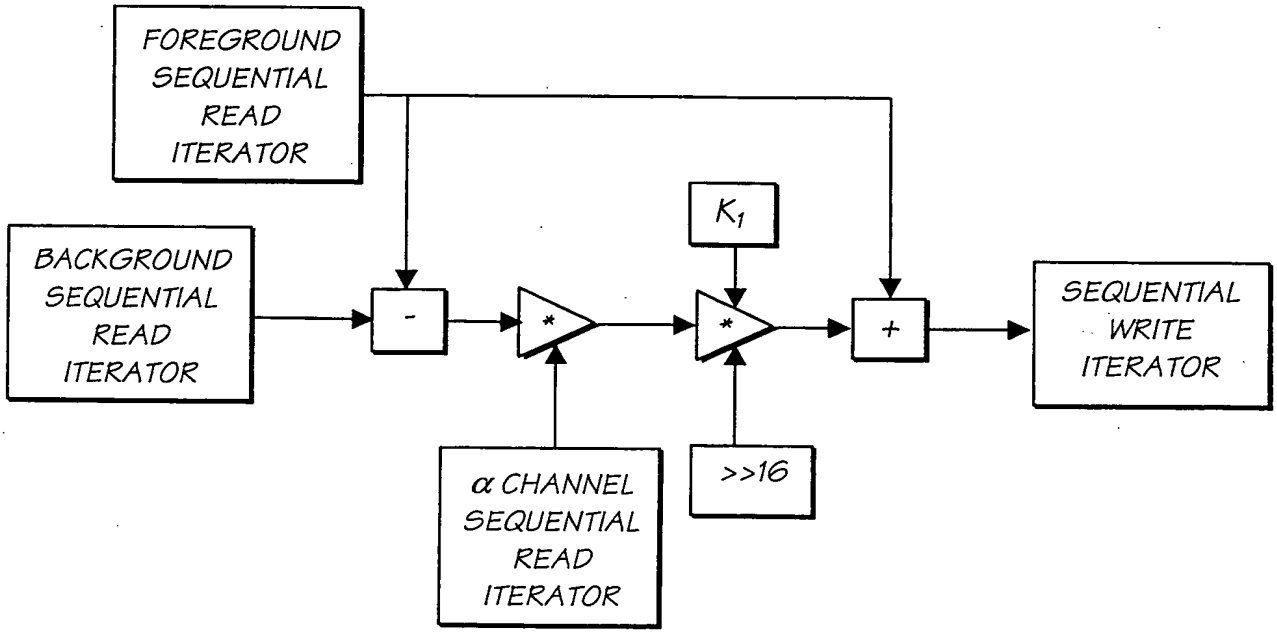

FOREGROUND SEQUENTIAL READ ITERATOR
BACKGROUND SEQUENTIAL READ ITERATOR
K1
-
*
*
+
SEQUENTIAL WRITE ITERATOR
α CHANNEL SEQUENTIAL READ ITERATOR
>>16

FIG. 84

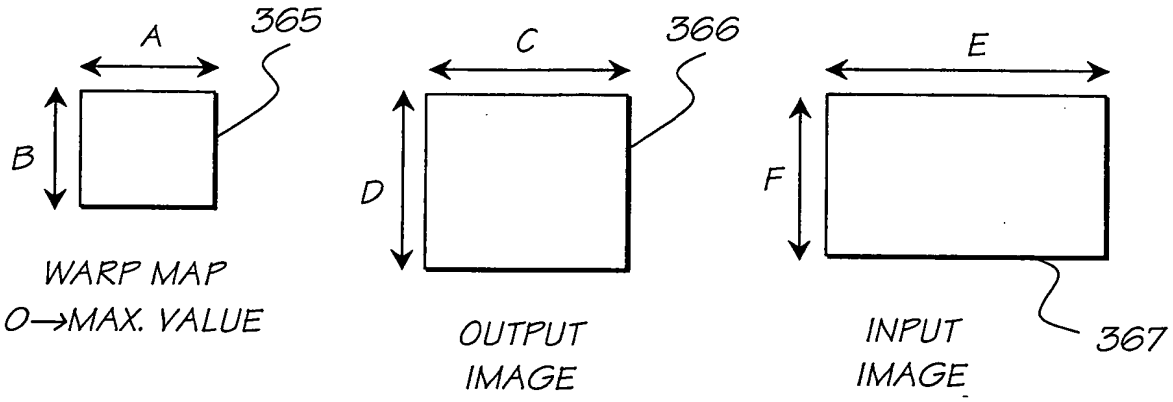


FIG. 85

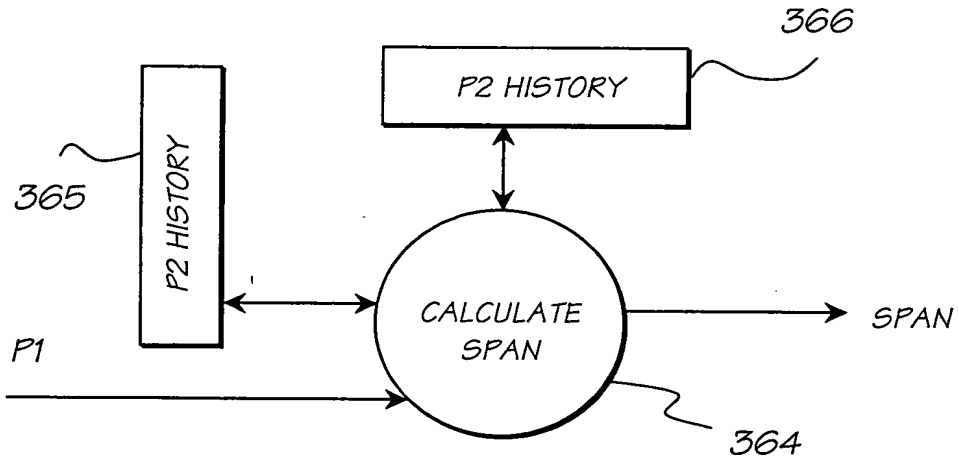


FIG. 86

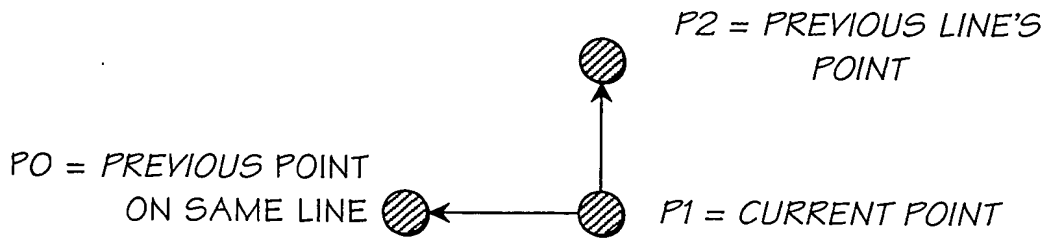


FIG. 88

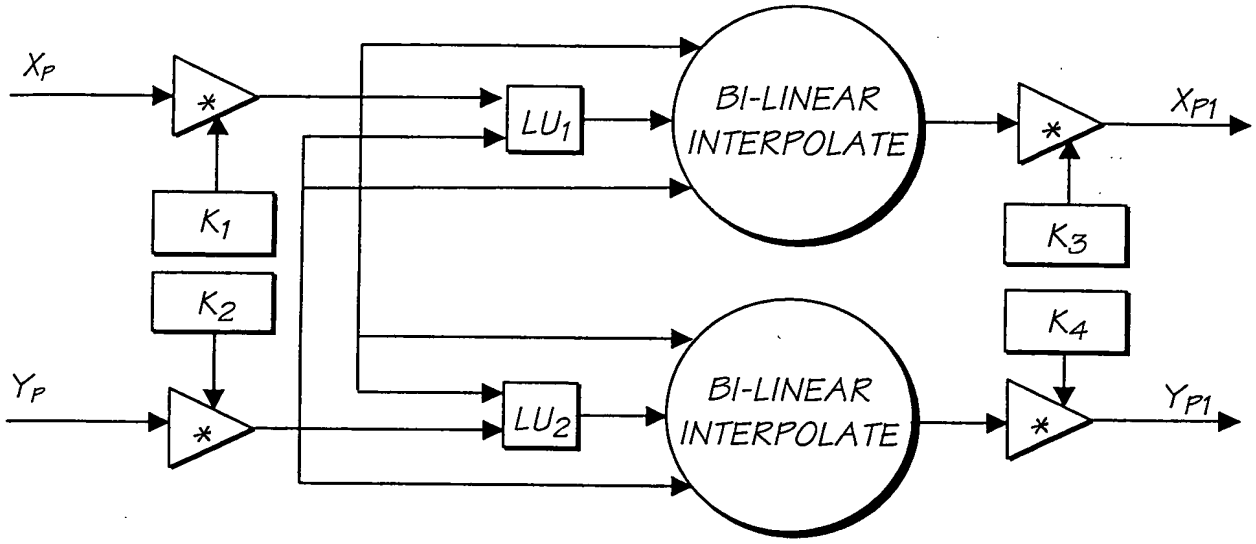

X_P
Y_P
K_1
K_2
LU_1
LU_2
BI-LINEAR INTERPOLATE
BI-LINEAR INTERPOLATE
K_3
K_4
X_{P1}
Y_{P1}

FIG. 87

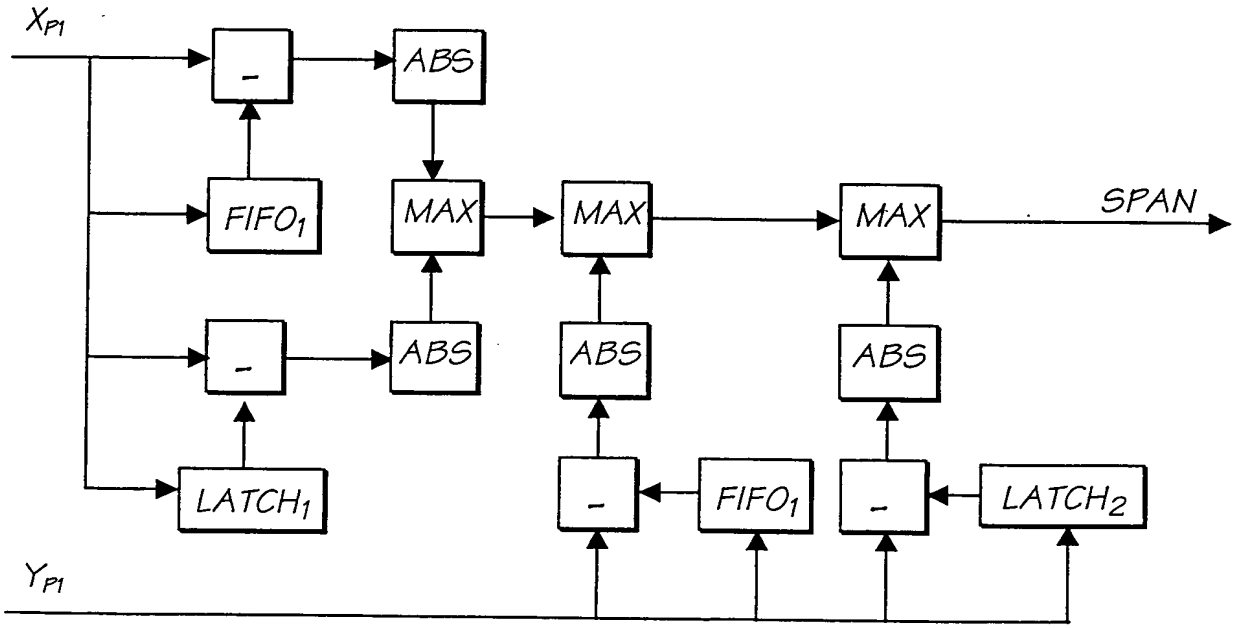

X_{P1}
Y_{P1}
ABS
ABS
ABS
ABS
$FIFO_1$
$FIFO_1$
MAX
MAX
MAX
SPAN
$LATCH_1$
$LATCH_2$

FIG. 89

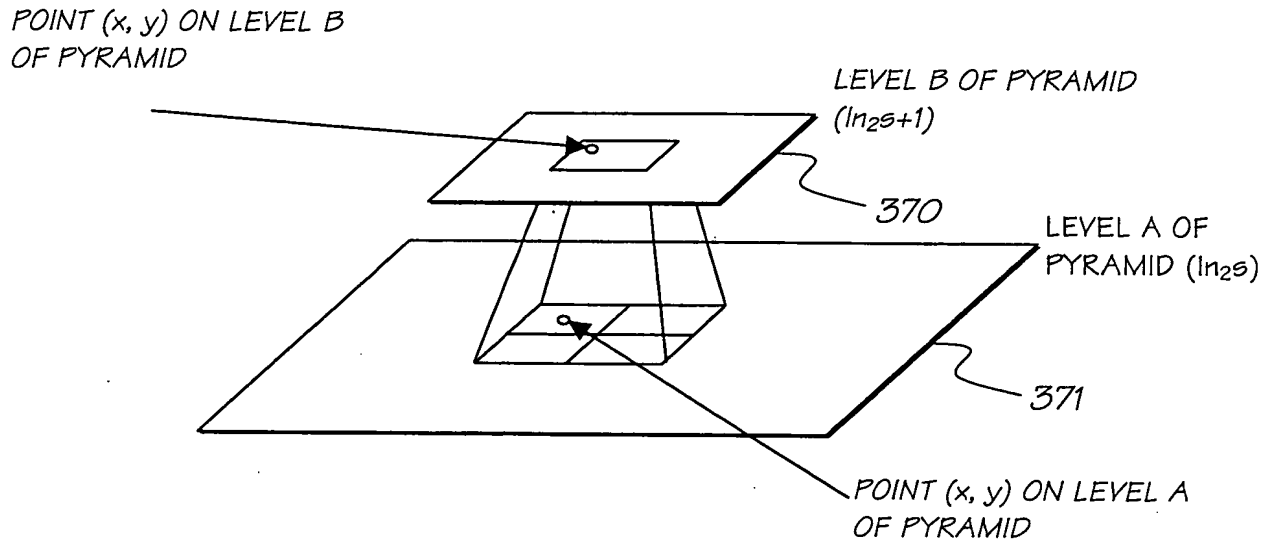

POINT (x, y) ON LEVEL B OF PYRAMID
LEVEL B OF PYRAMID ($ln_2 s+1$)
370
LEVEL A OF PYRAMID ($ln_2 s$)
371
POINT (x, y) ON LEVEL A OF PYRAMID

FIG. 90

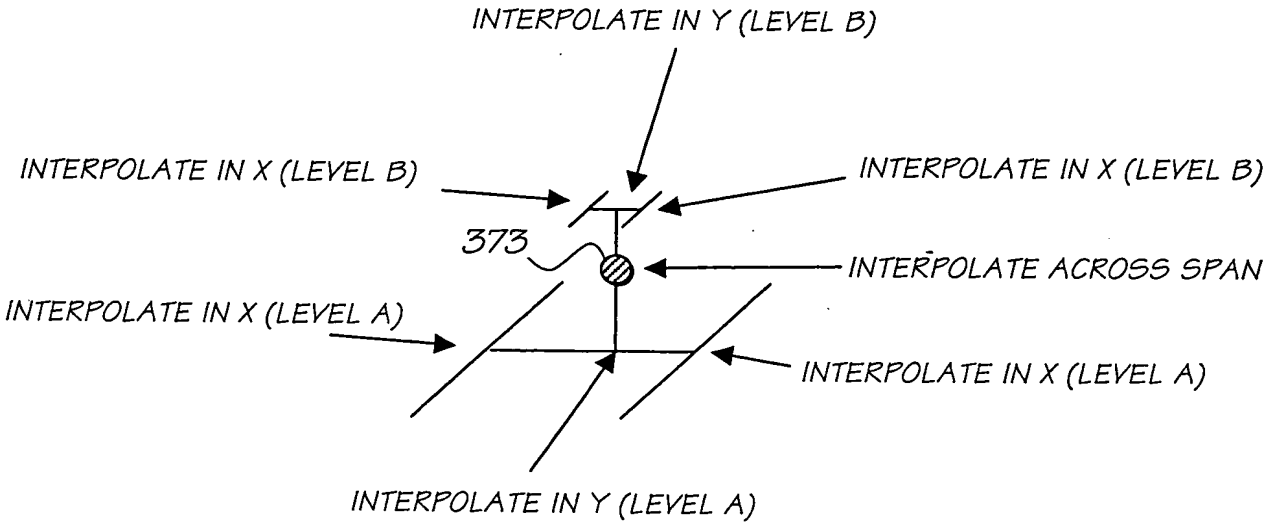

INTERPOLATE IN Y (LEVEL B)
INTERPOLATE IN X (LEVEL B)
INTERPOLATE IN X (LEVEL B)
373
INTERPOLATE ACROSS SPAN
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN Y (LEVEL A)

FIG. 91

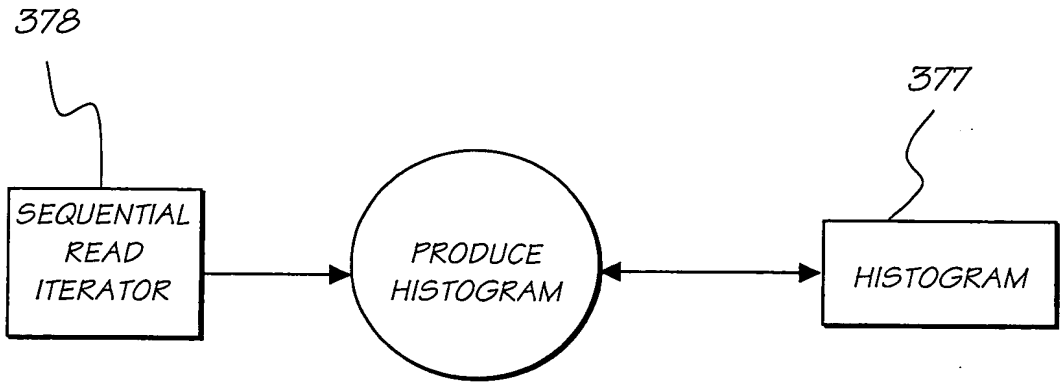

378
377
SEQUENTIAL READ ITERATOR
PRODUCE HISTOGRAM
HISTOGRAM

FIG. 92

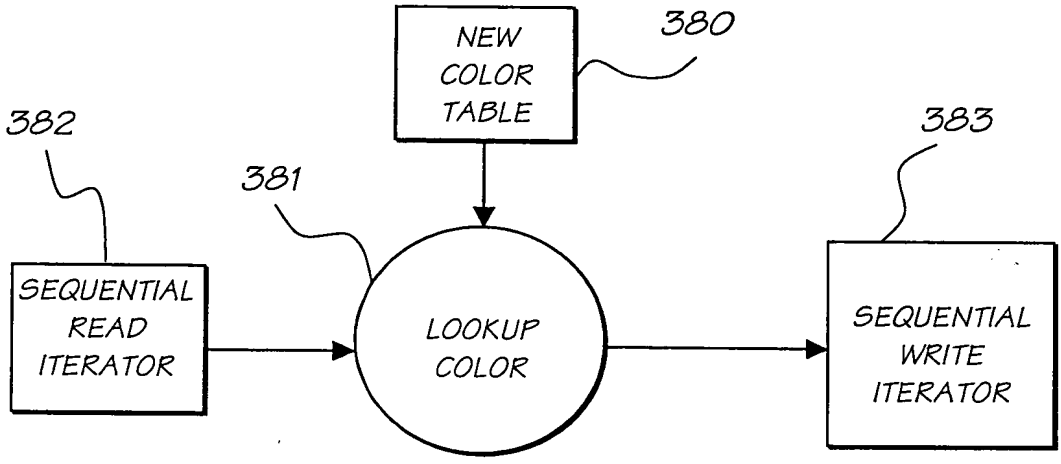

NEW COLOR TABLE
380
382
383
381
SEQUENTIAL READ ITERATOR
LOOKUP COLOR
SEQUENTIAL WRITE ITERATOR

FIG. 93

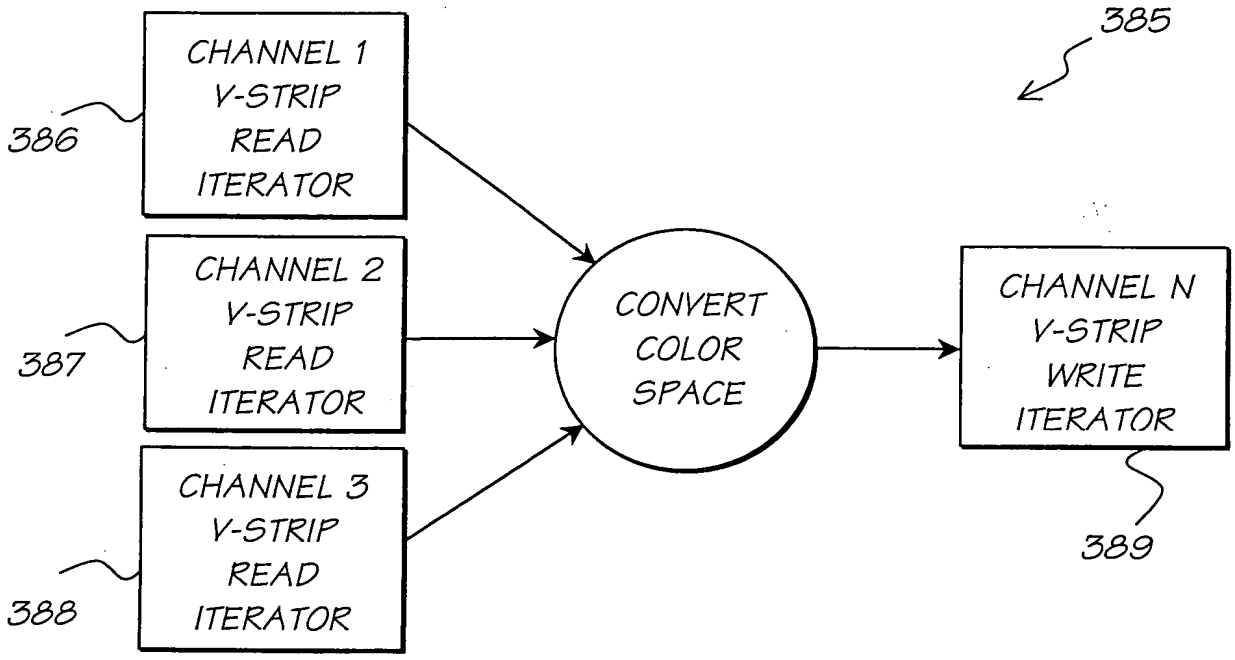


FIG. 94

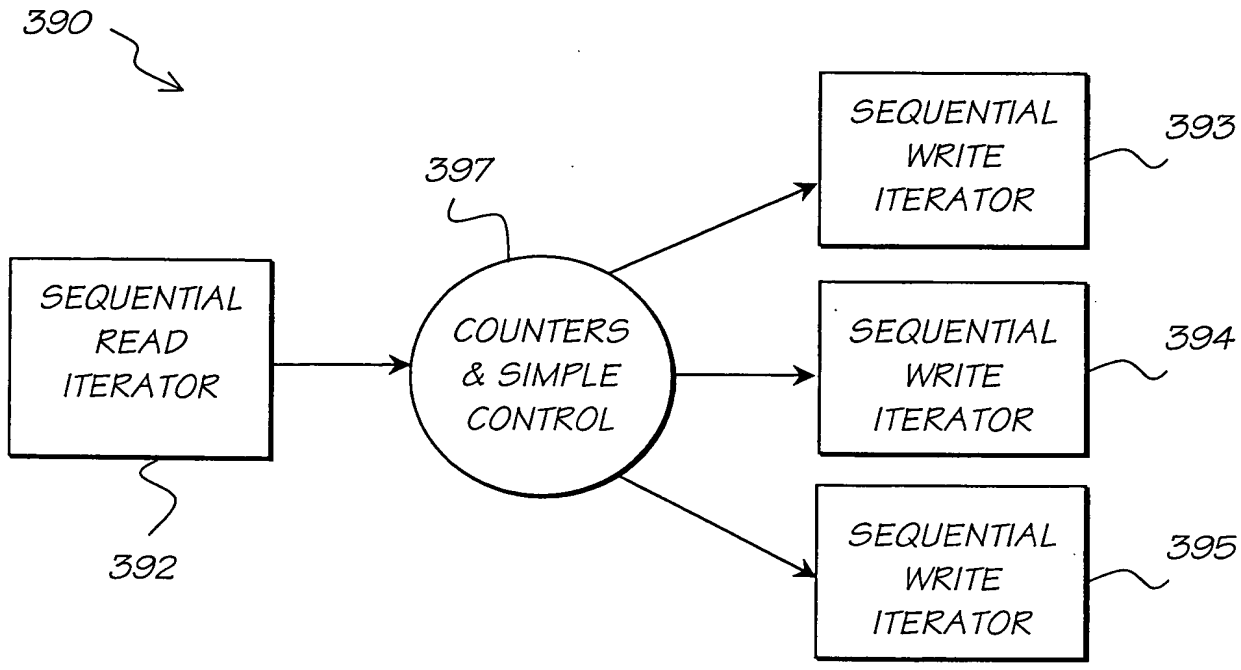


FIG. 101

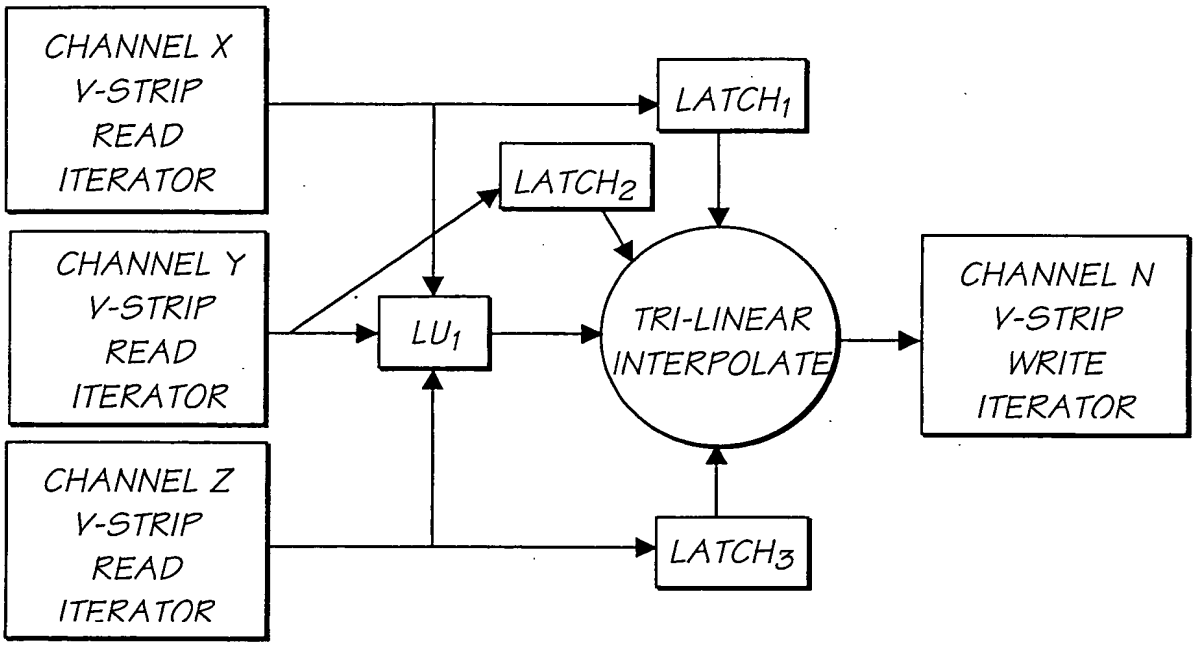

CHANNEL X V-STRIP READ ITERATOR
CHANNEL Y V-STRIP READ ITERATOR
CHANNEL Z V-STRIP READ ITERATOR
LATCH1
LATCH2
LU1
TRI-LINEAR INTERPOLATE
LATCH3
CHANNEL N V-STRIP WRITE ITERATOR

FIG. 95

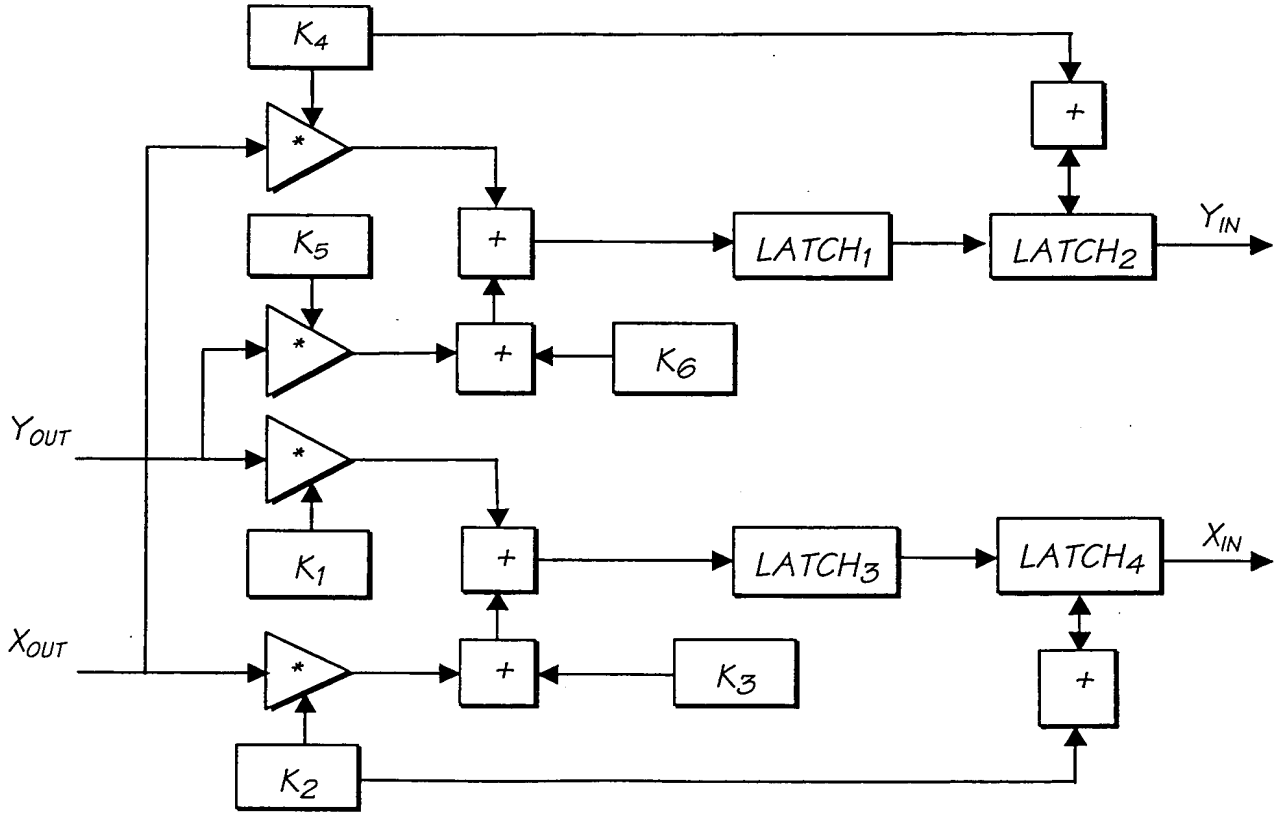

K4
K5
K6
K1
K2
K3
*
+
LATCH1
LATCH2
LATCH3
LATCH4
YIN
XIN
YOUT
XOUT

FIG. 96

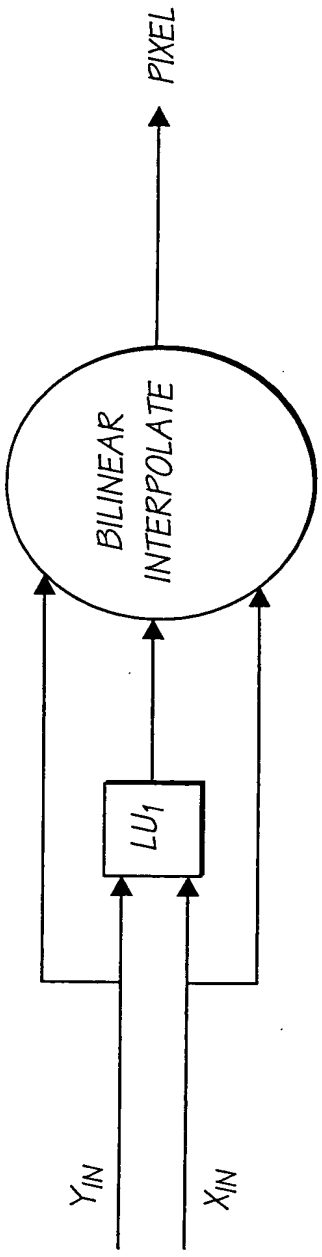

PIXEL
BILINEAR INTERPOLATE
LU1
YIN
XIN

FIG. 97

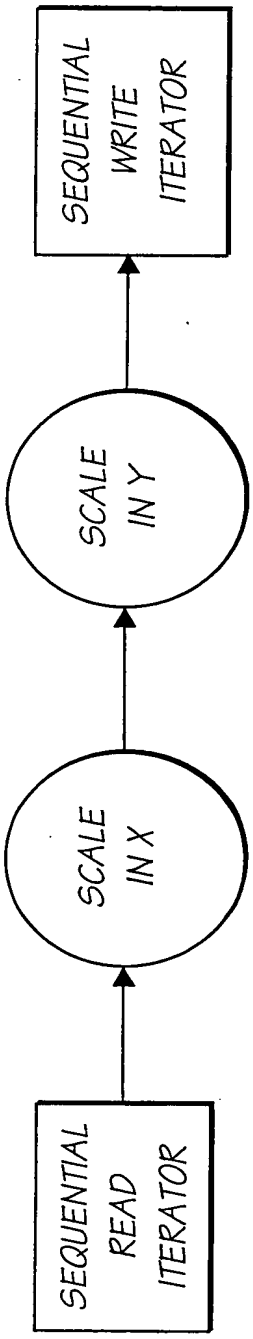

SEQUENTIAL WRITE ITERATOR
SCALE IN Y
SCALE IN X
SEQUENTIAL READ ITERATOR

FIG. 98

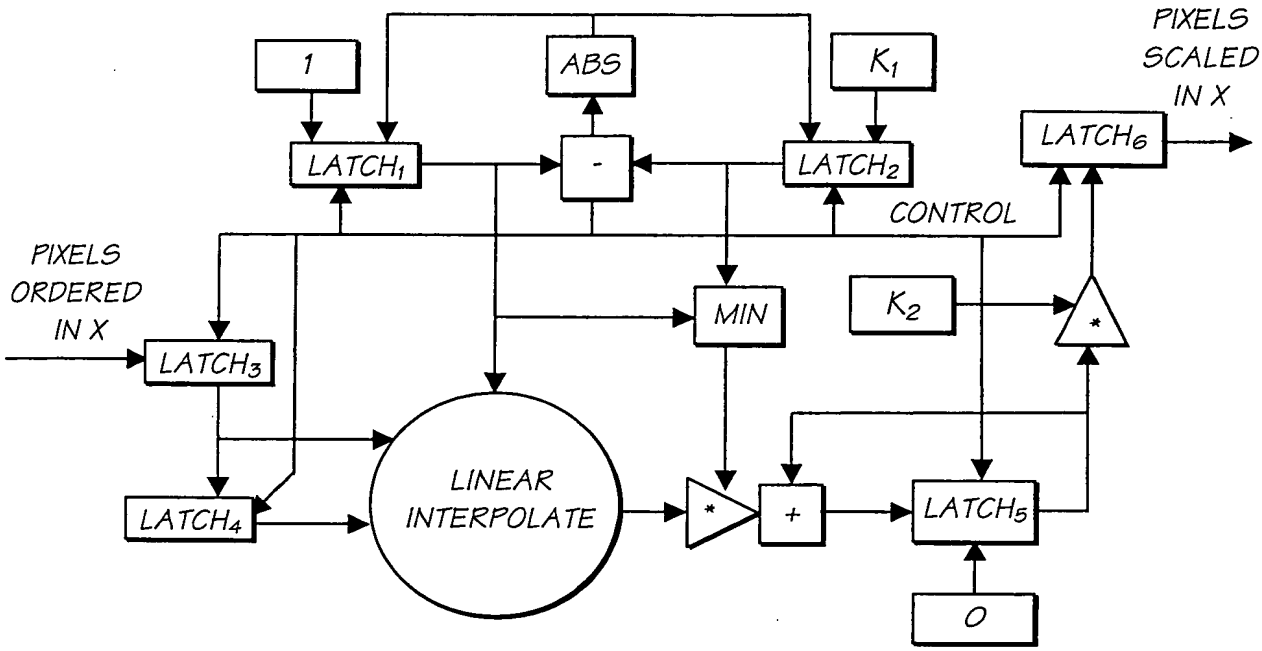


FIG. 99

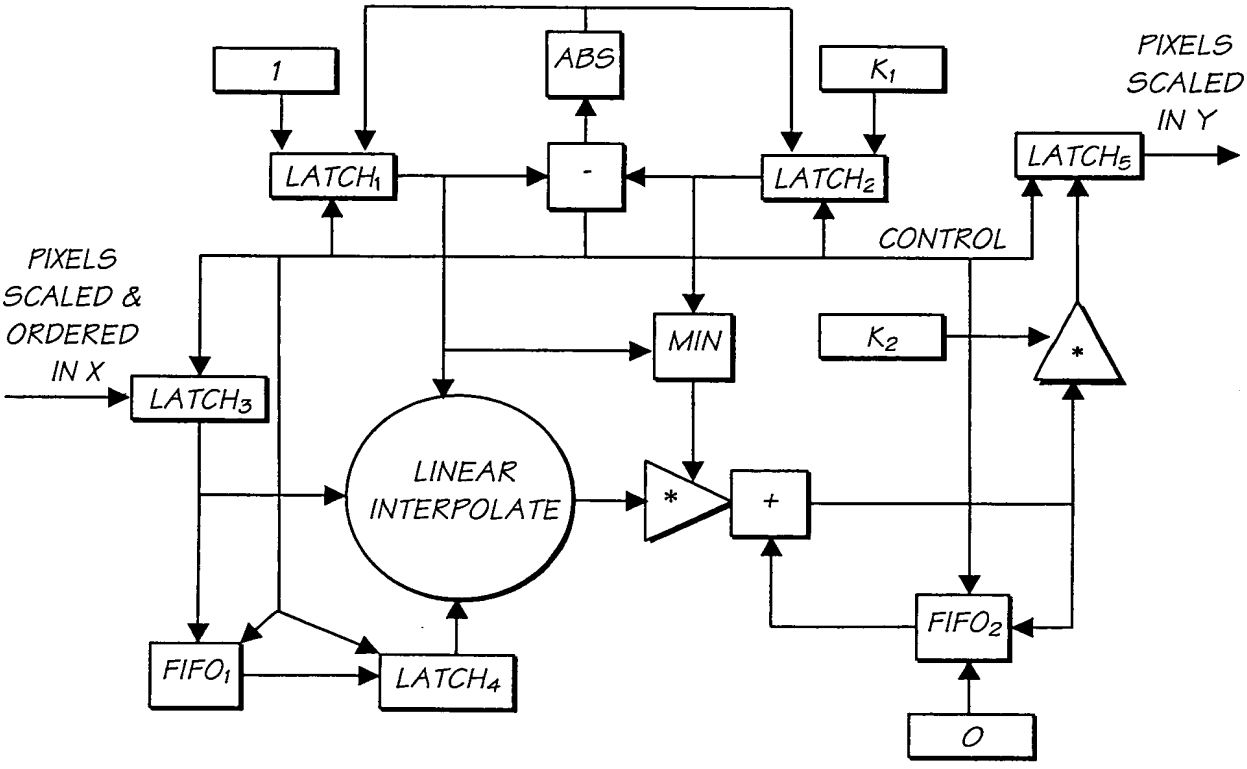


FIG. 100

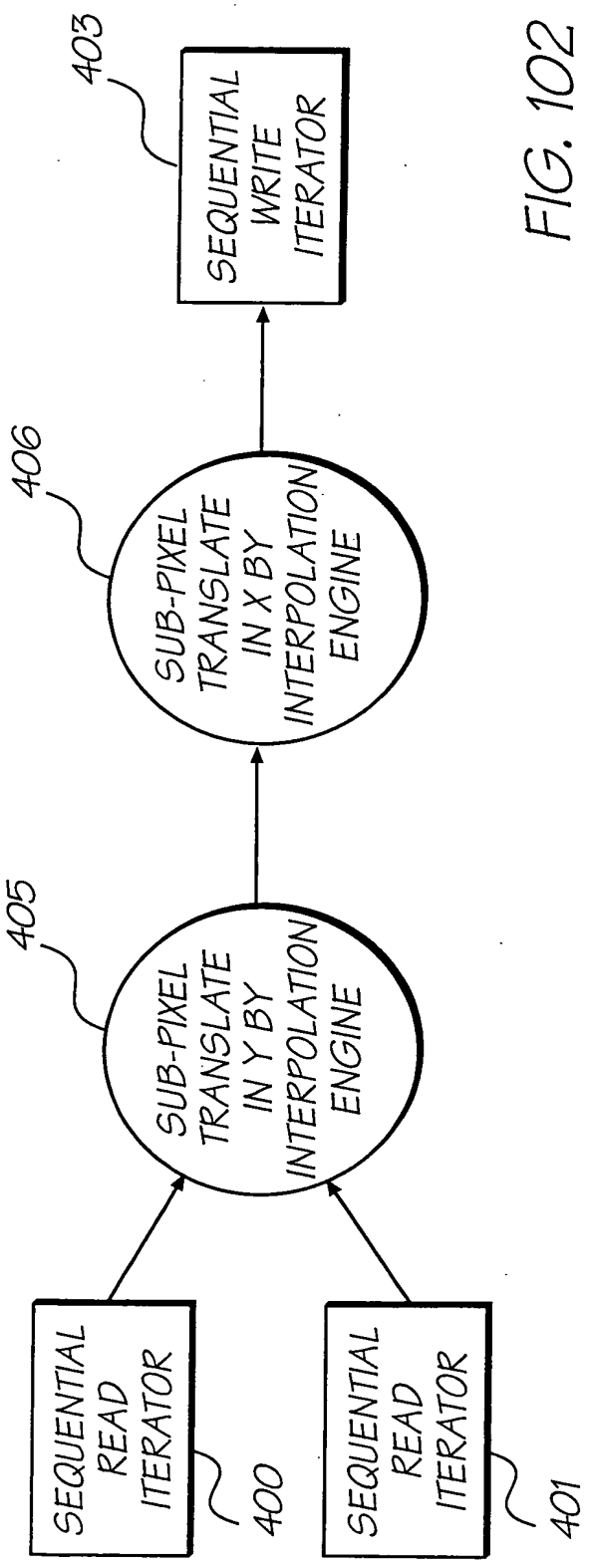


FIG. 102

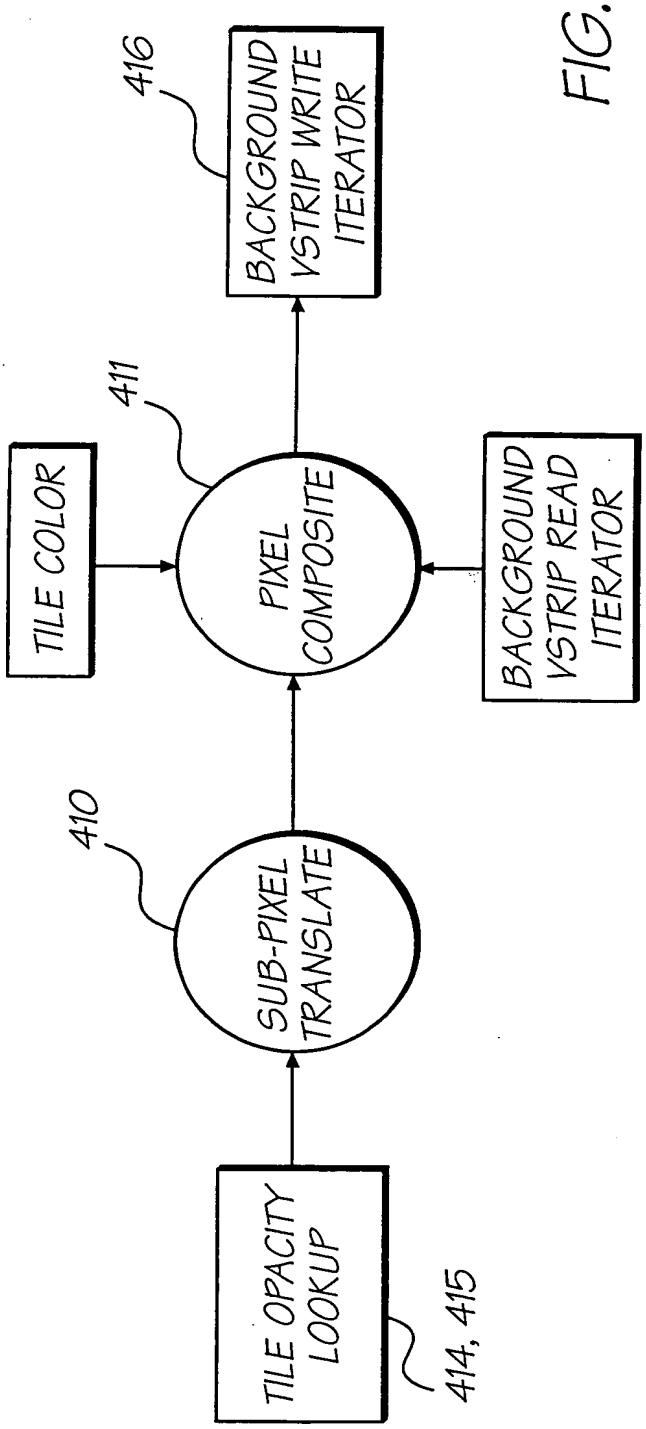


FIG. 103

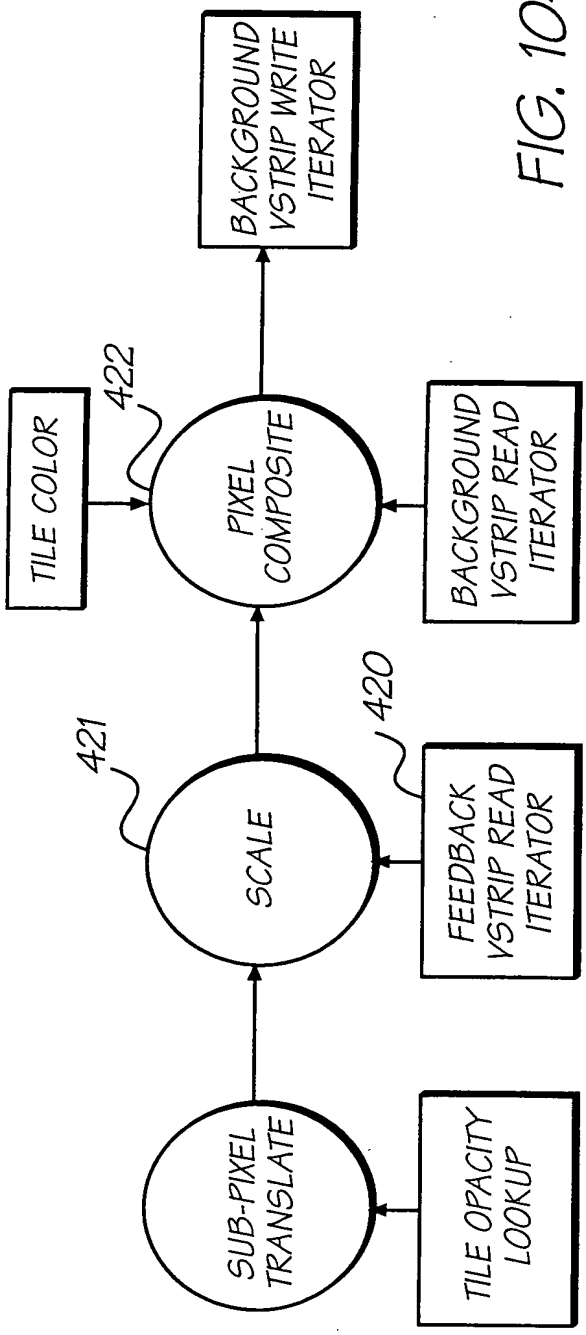


FIG. 104

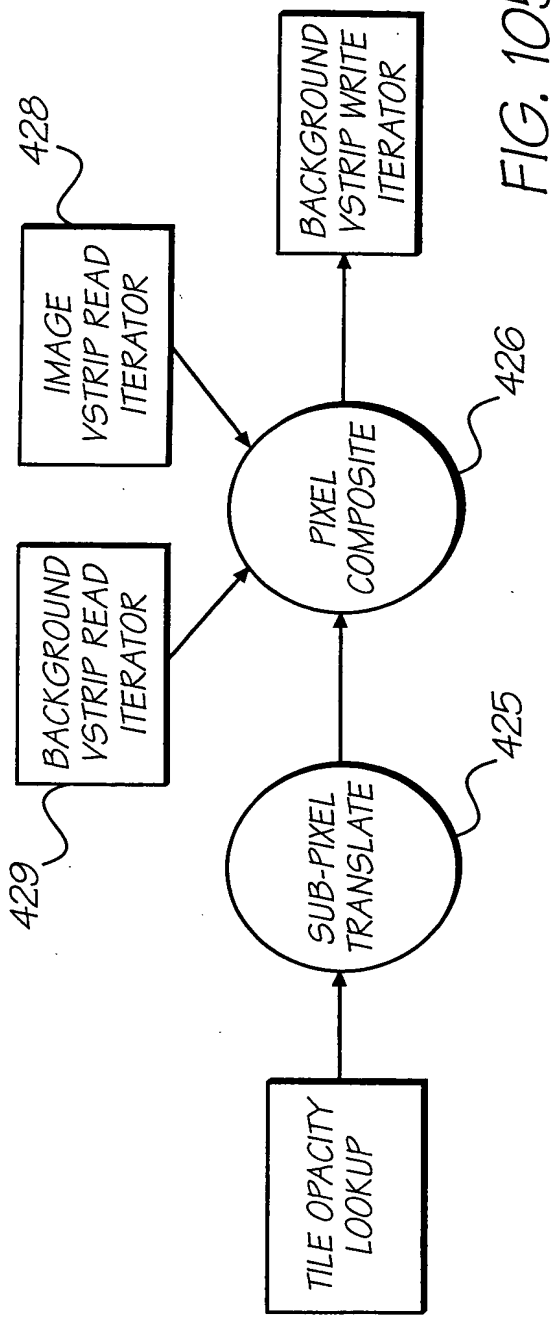


FIG. 105

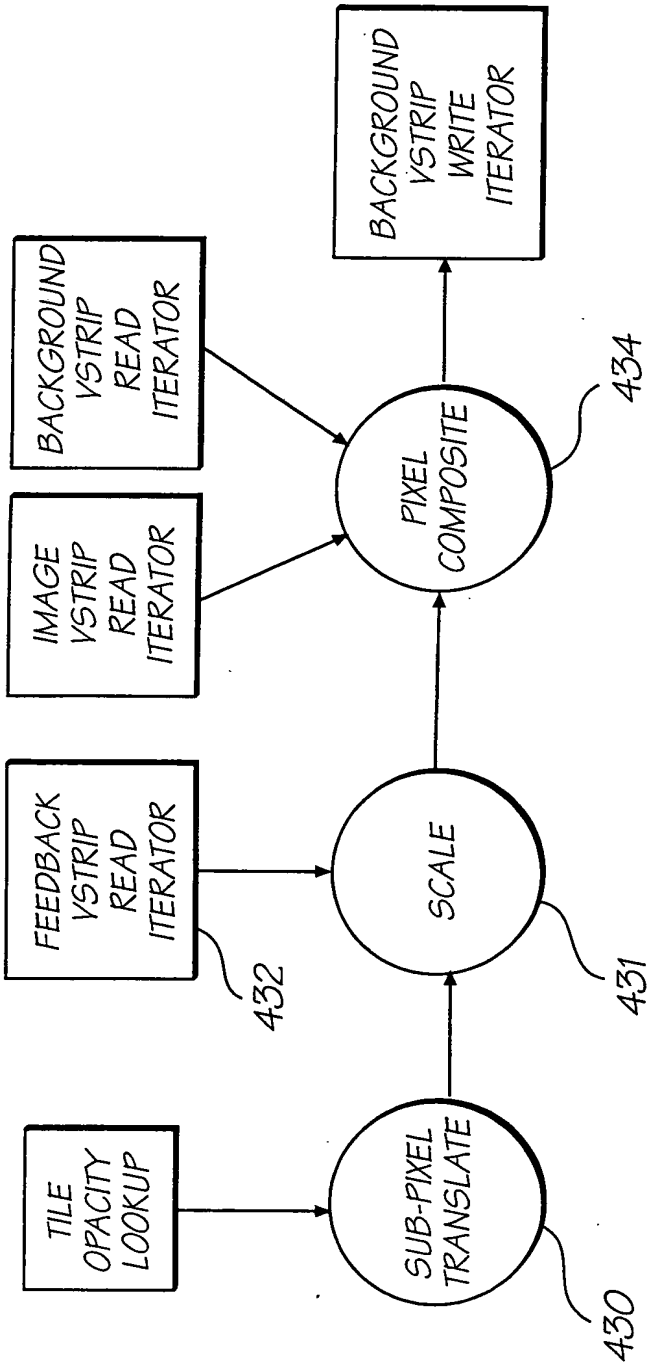


FIG. 106

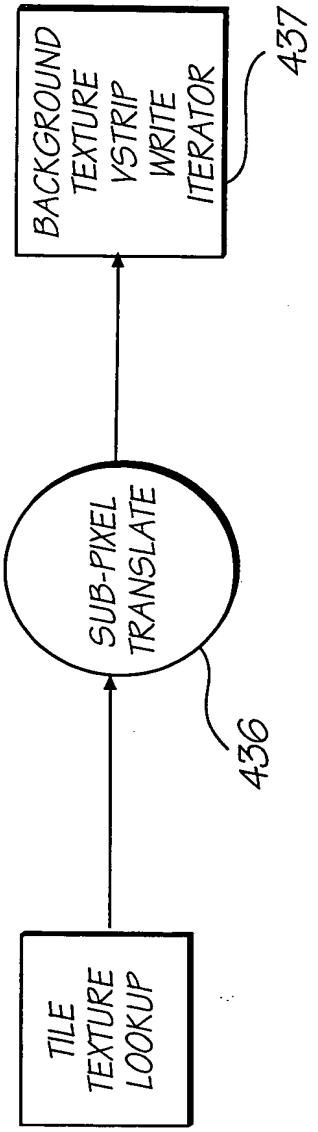


FIG. 107

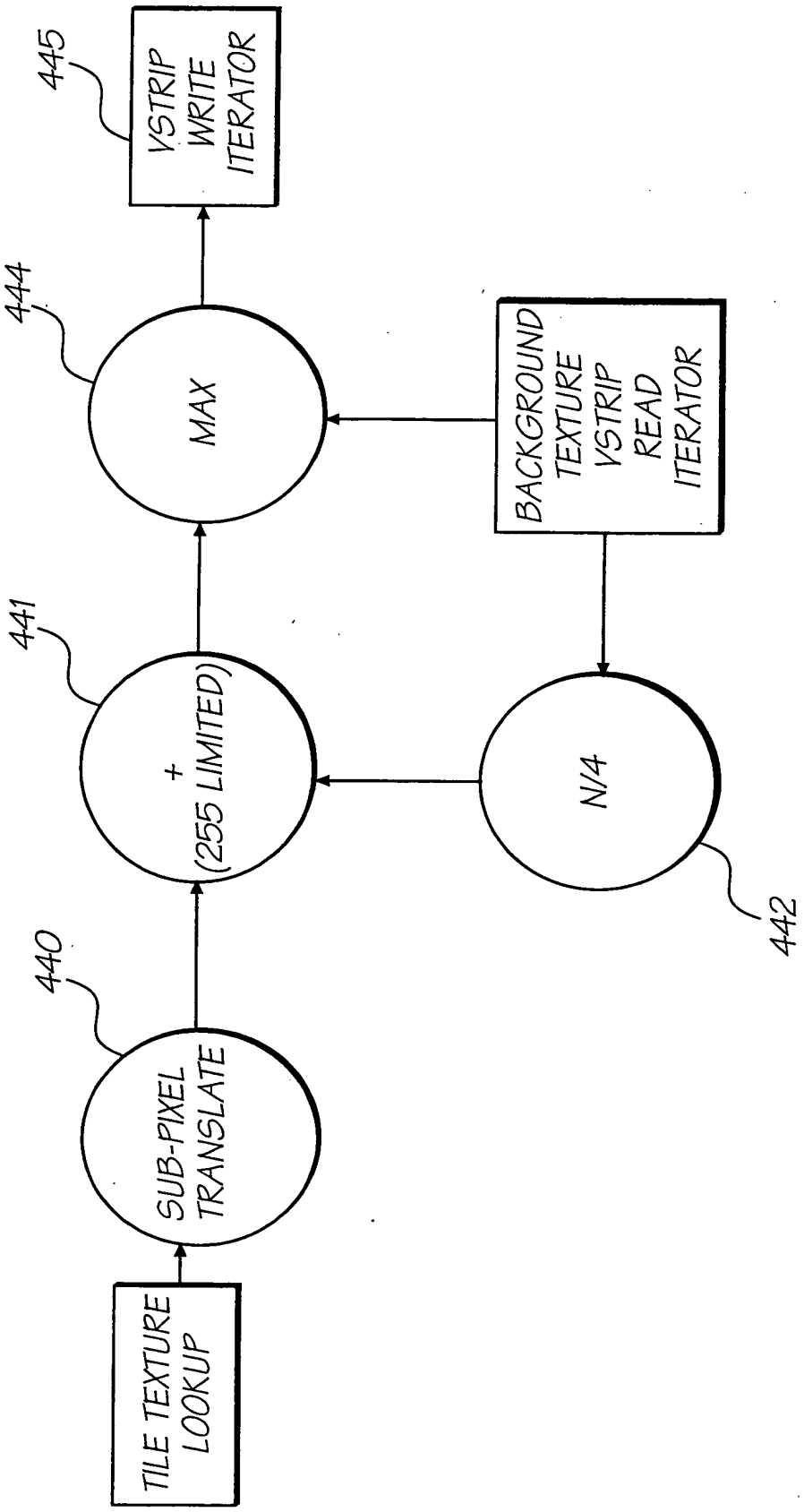

TILE TEXTURE LOOKUP
SUB-PIXEL TRANSLATE
440
+ (255 LIMITED)
441
N/4
442
BACKGROUND TEXTURE VSTRIP READ ITERATOR
MAX
444
VSTRIP WRITE ITERATOR
445

FIG. 108

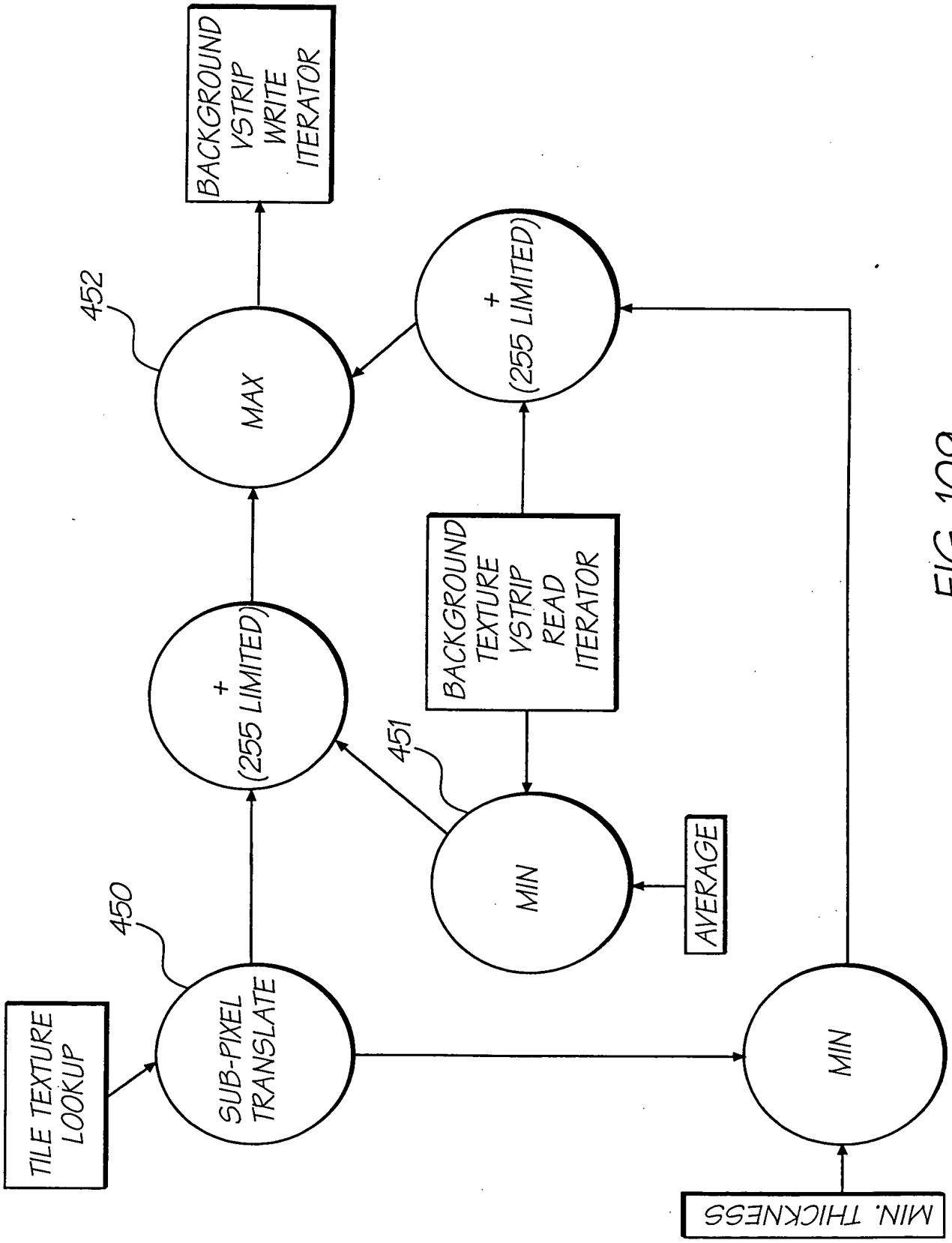

TILE TEXTURE LOOKUP
SUB-PIXEL TRANSLATE
450
MIN
MIN. THICKNESS
AVERAGE
MIN
451
+ (255 LIMITED)
BACKGROUND TEXTURE YSTRIP READ ITERATOR
+ (255 LIMITED)
MAX
452
BACKGROUND YSTRIP WRITE ITERATOR

FIG. 109

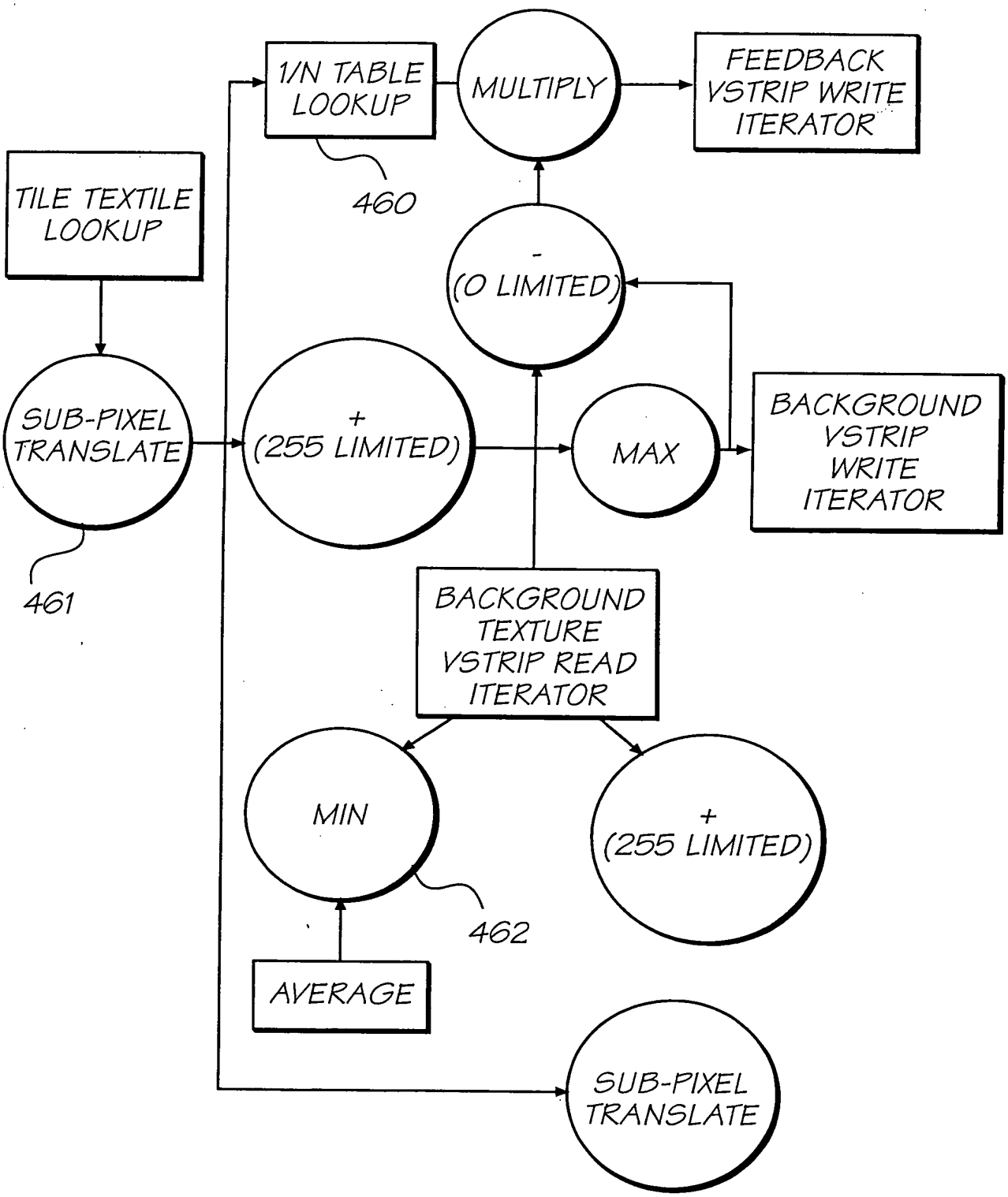
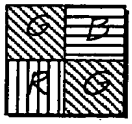

1/N TABLE LOOKUP
MULTIPLY
FEEDBACK VSTRIP WRITE ITERATOR
460
TILE TEXTILE LOOKUP
- (0 LIMITED)
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
MAX
BACKGROUND VSTRIP WRITE ITERATOR
461
BACKGROUND TEXTURE VSTRIP READ ITERATOR
MIN
+ (255 LIMITED)
462
AVERAGE
SUB-PIXEL TRANSLATE

FIG. 110

2X2 PIXEL BLOCK,
0 DEGREES

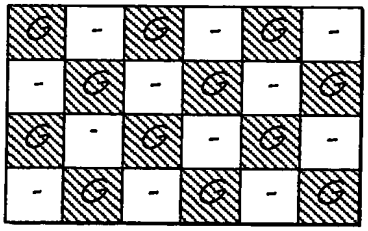
2X2 PIXEL BLOCK,
90 DEGREES



2X2 PIXEL BLOCK,
180 DEGREES

2X2 PIXEL BLOCK,
270 DEGREES

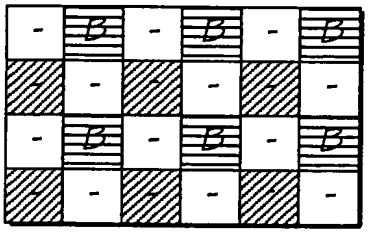
FIG. 111



LINEAR INTERPOLATED PIXELS

ACTUAL PIXELS (NOT INTERPOLATED)

FIG. 112

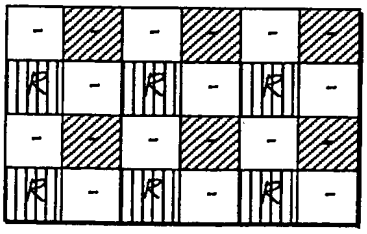


LINEAR INTERPOLATED PIXELS

BI-LINEAR INTERPOLATED PIXELS

ACTUAL PIXELS (NOT INTERPOLATED)

FIG. 113



LINEAR INTERPOLATED PIXELS

BI-LINEAR INTERPOLATED PIXELS

ACTUAL PIXELS (NOT INTERPOLATED)

FIG. 114

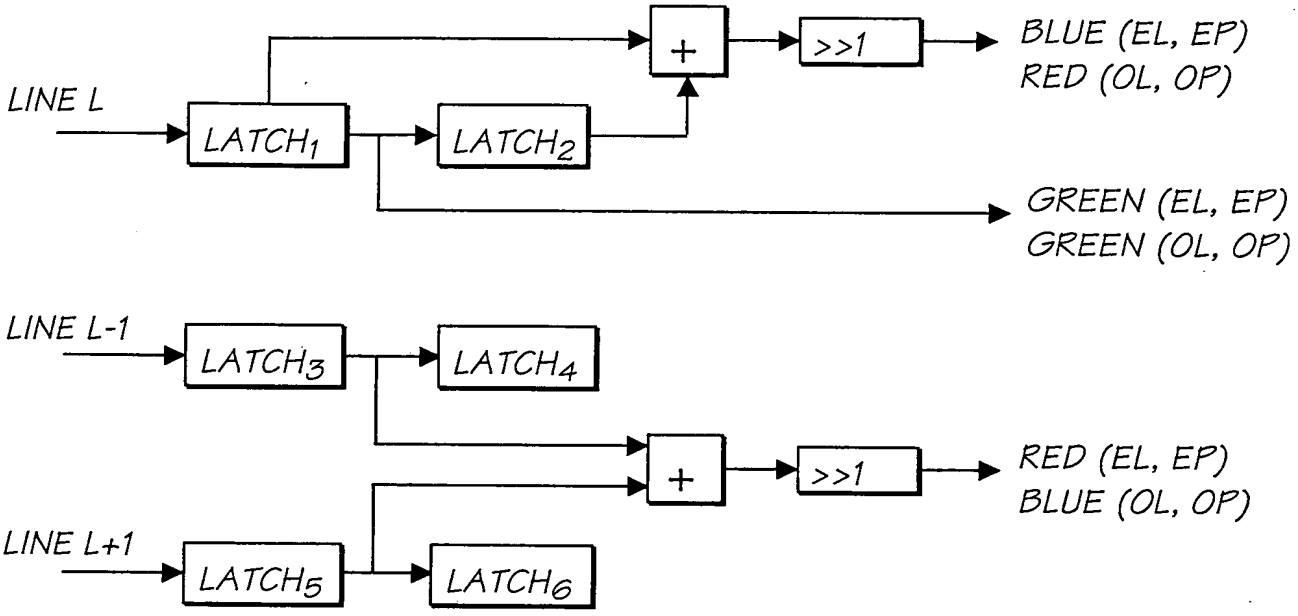


FIG. 115

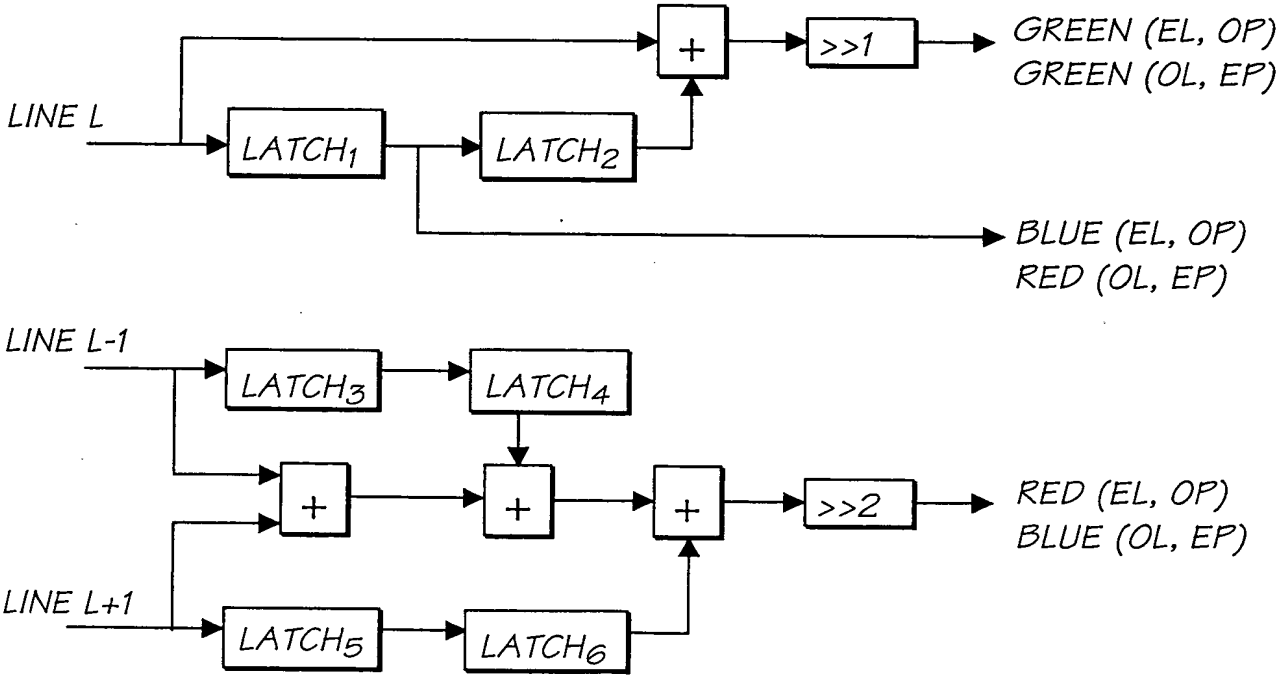


FIG. 116

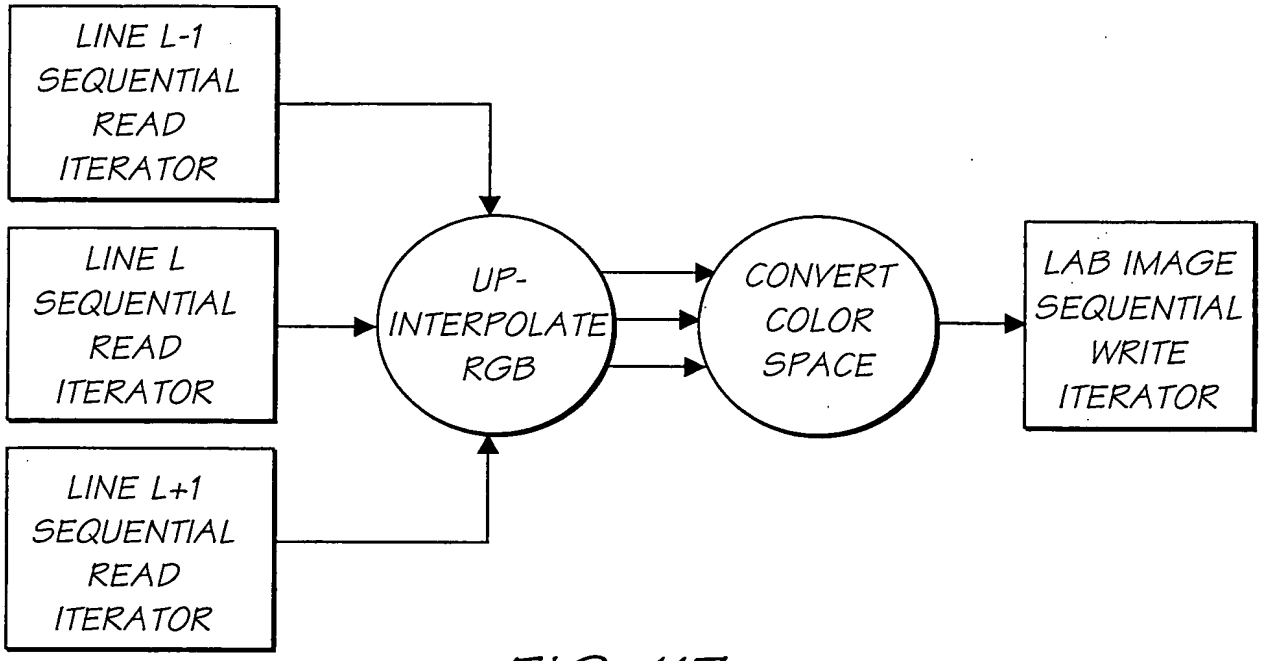


FIG. 117

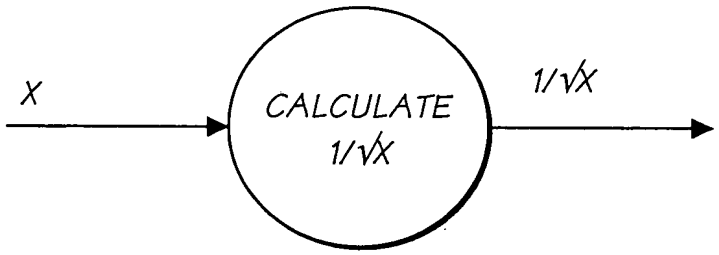


FIG. 118

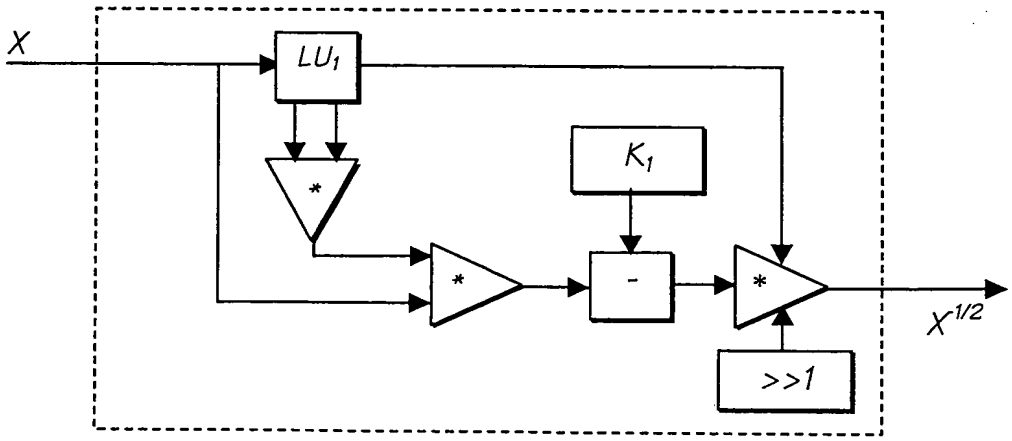


FIG. 119

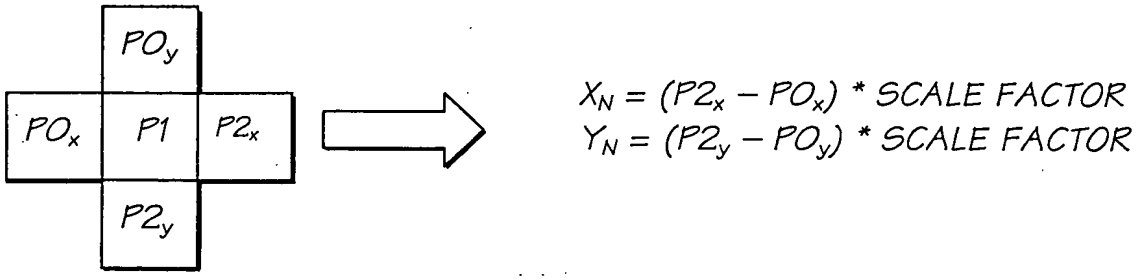


FIG. 120

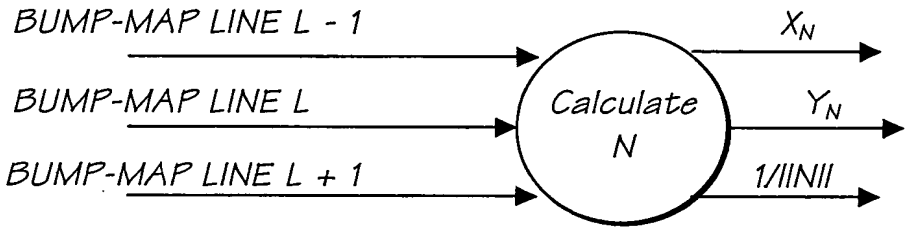


FIG. 121

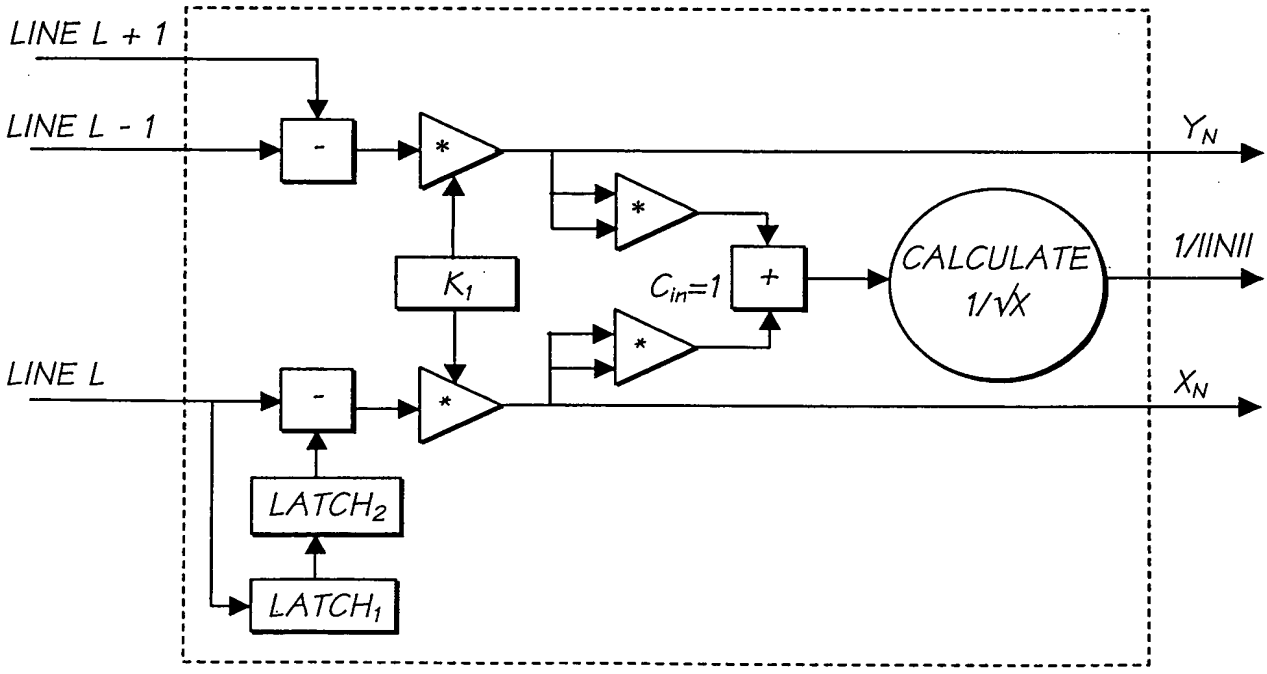


FIG. 122

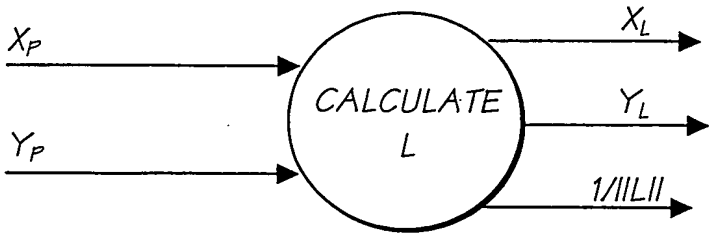


FIG. 123

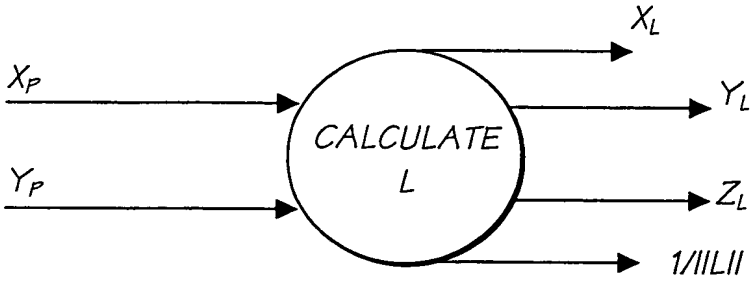


FIG. 124

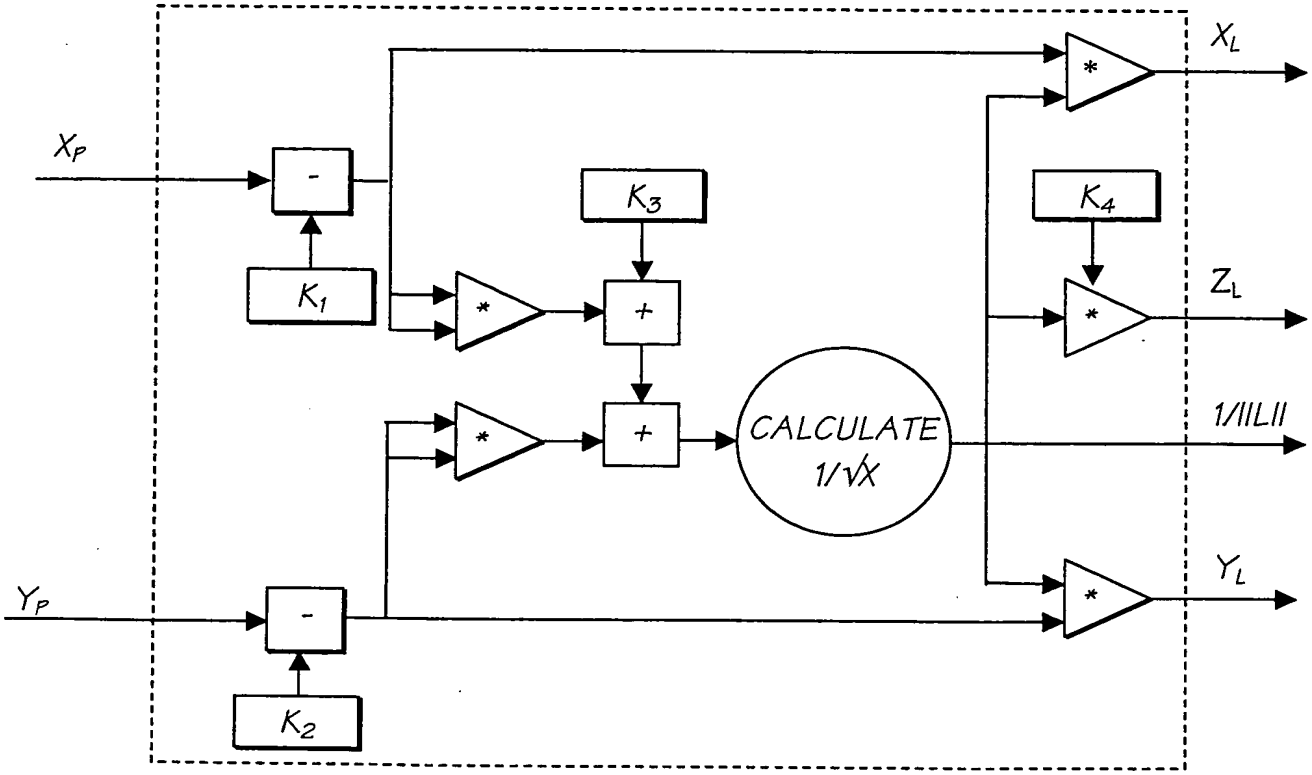


FIG. 125

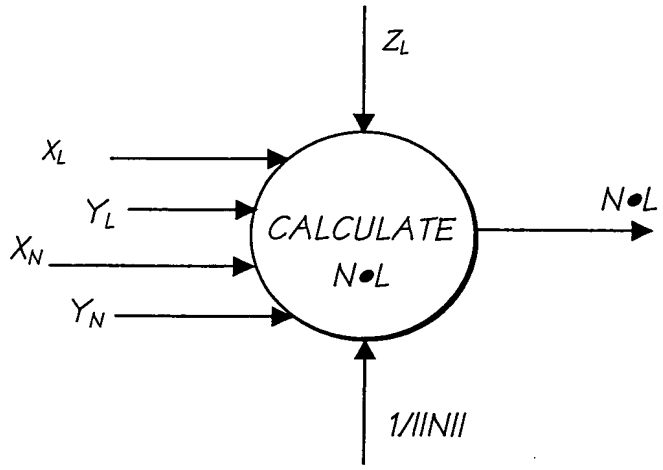


FIG. 126

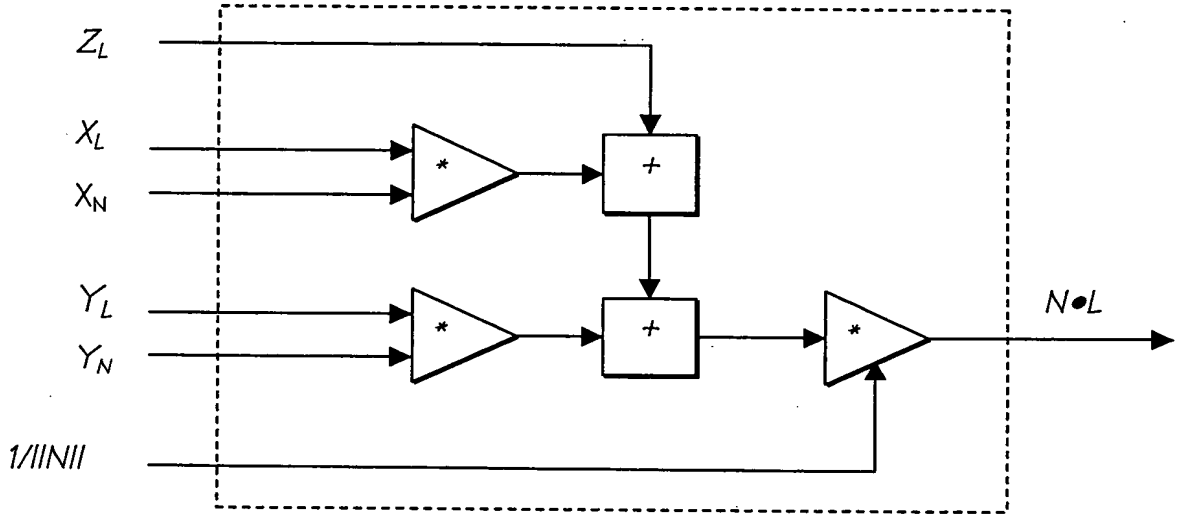


FIG. 127

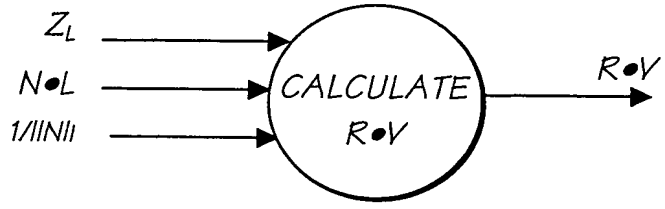


FIG. 128

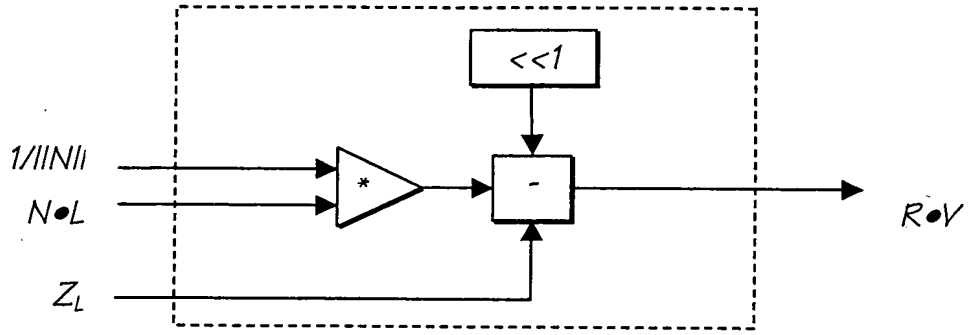


FIG. 129

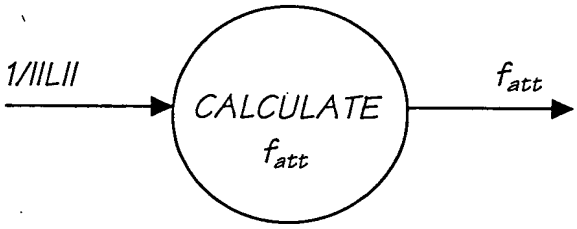


FIG. 130

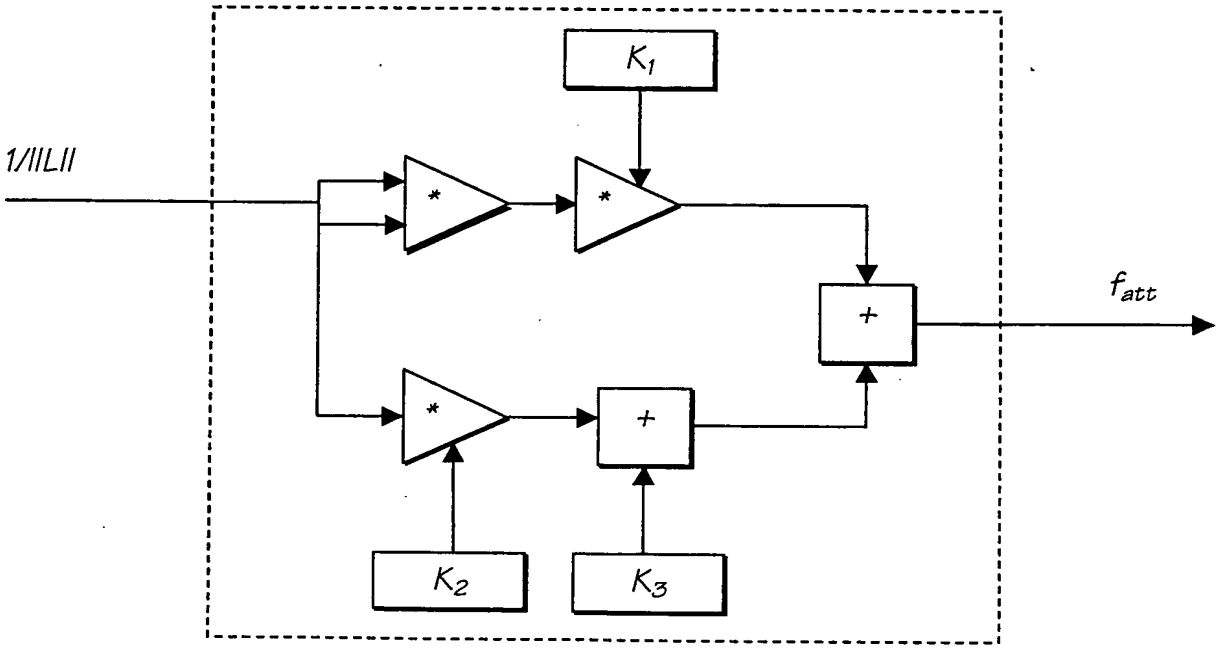


FIG. 131

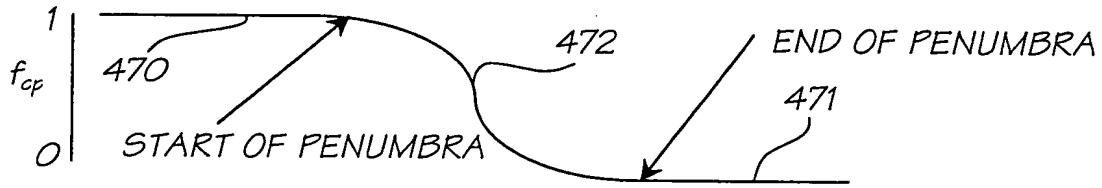


FIG. 132

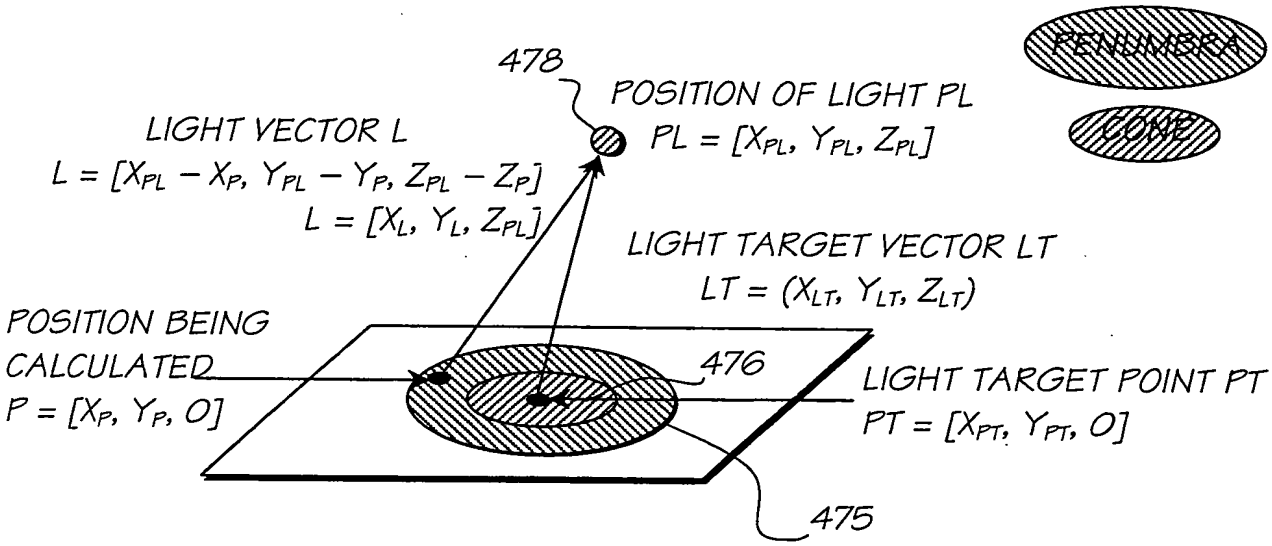


FIG. 133

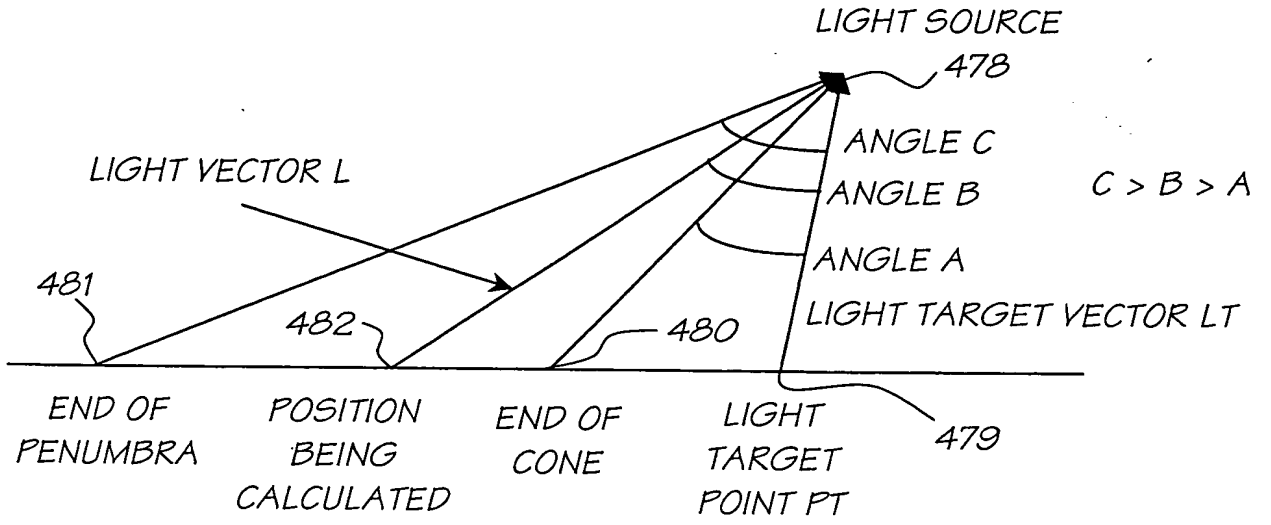


FIG. 134

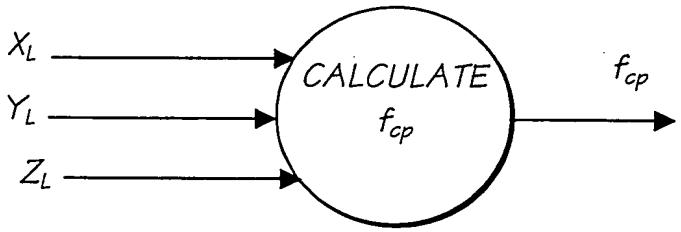


FIG. 135

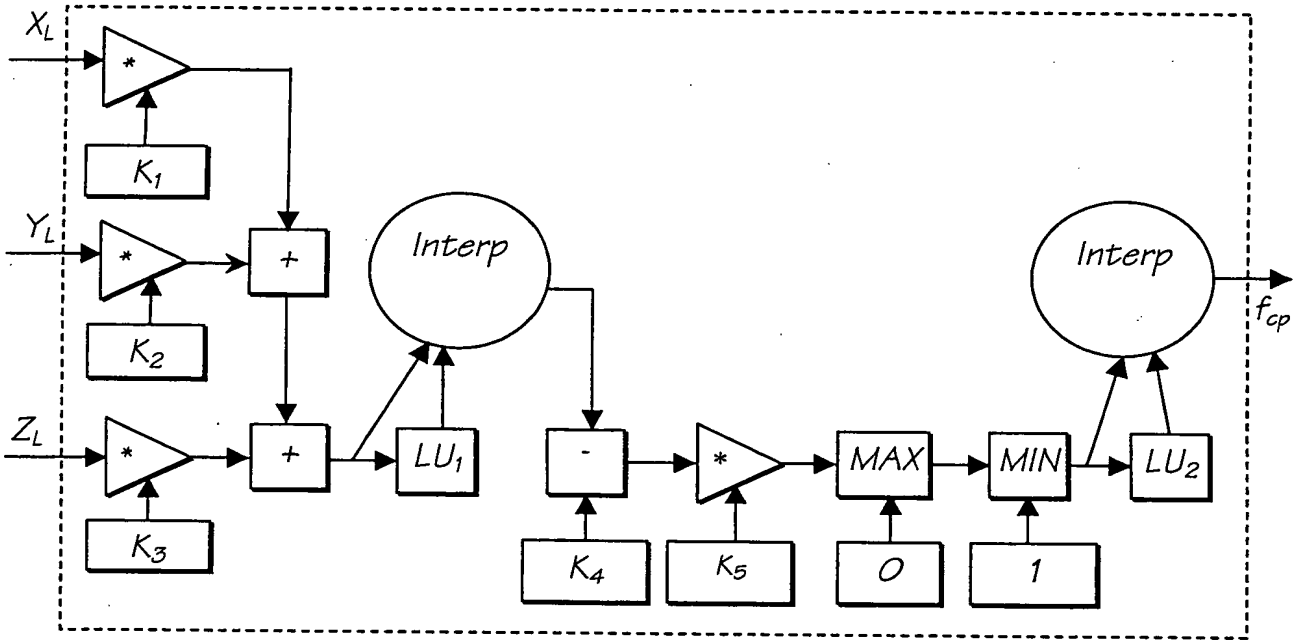


FIG. 136

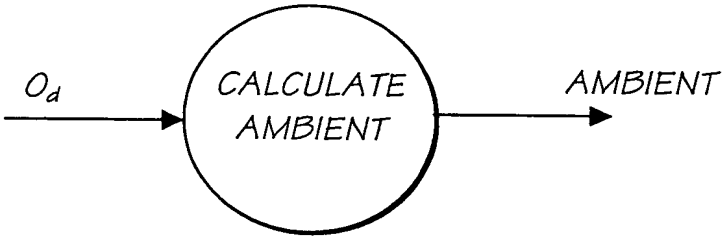


FIG. 137

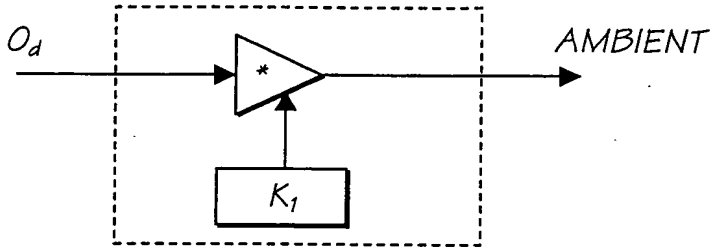


FIG. 138

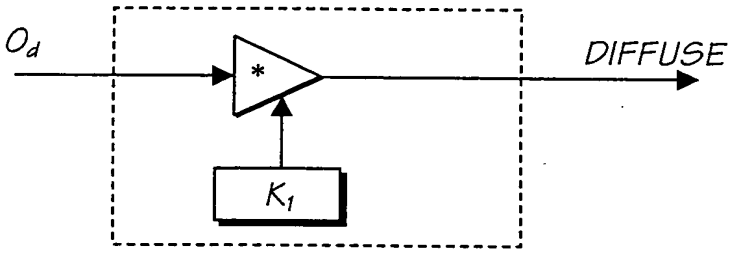


FIG. 139

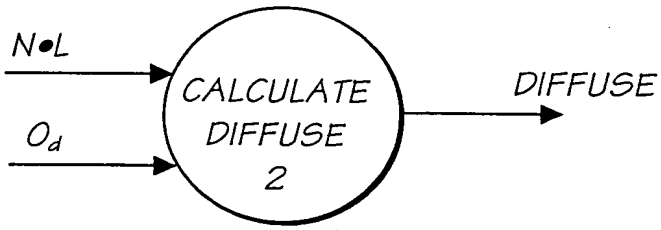


FIG. 140

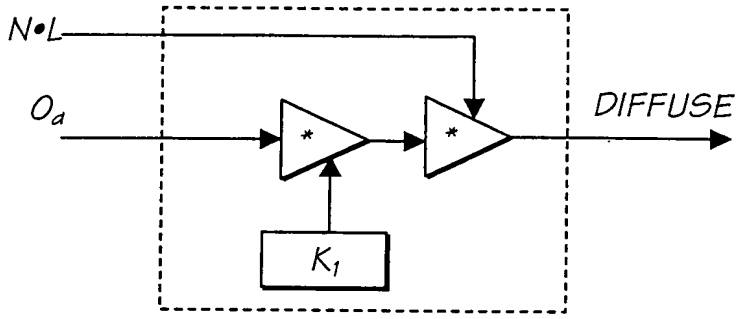


FIG. 141

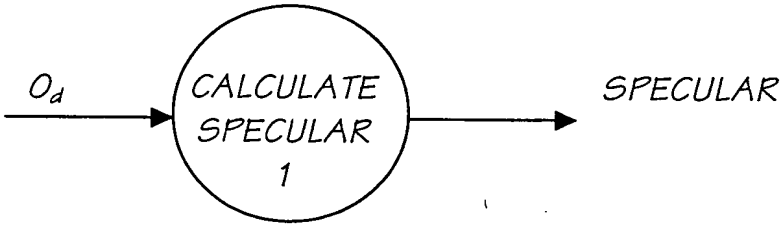


FIG. 142

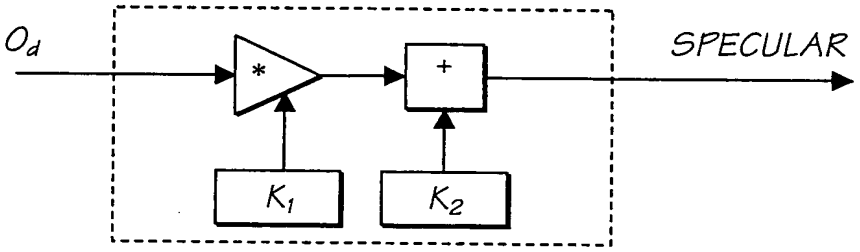


FIG. 143

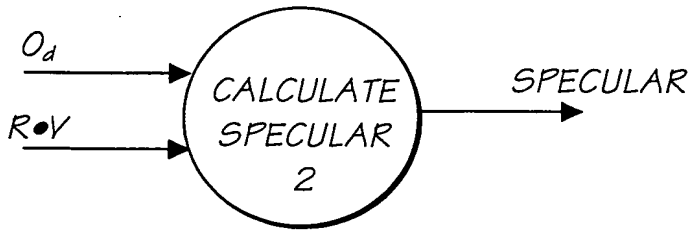


FIG. 144

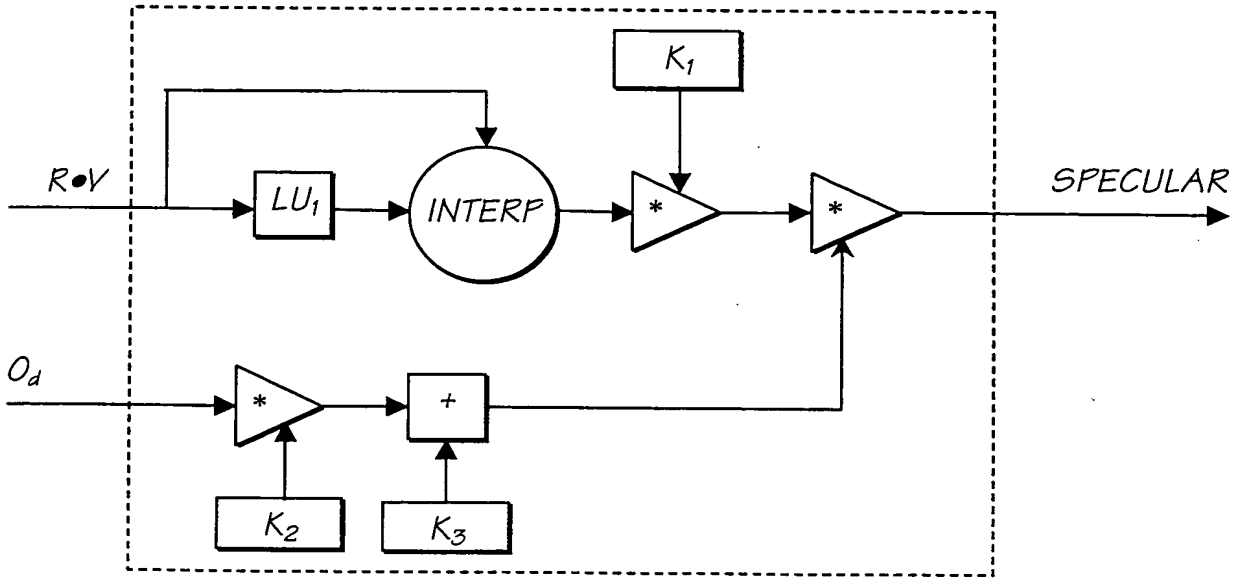


FIG. 145

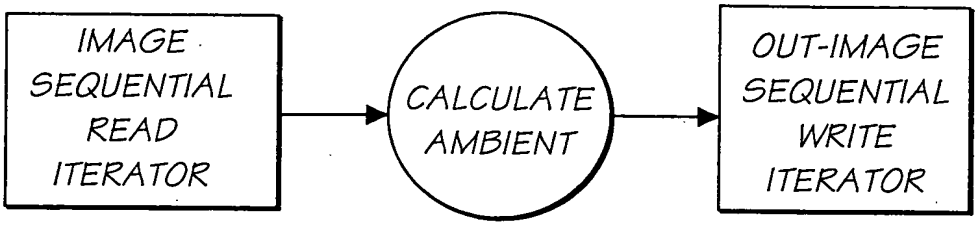


FIG. 146

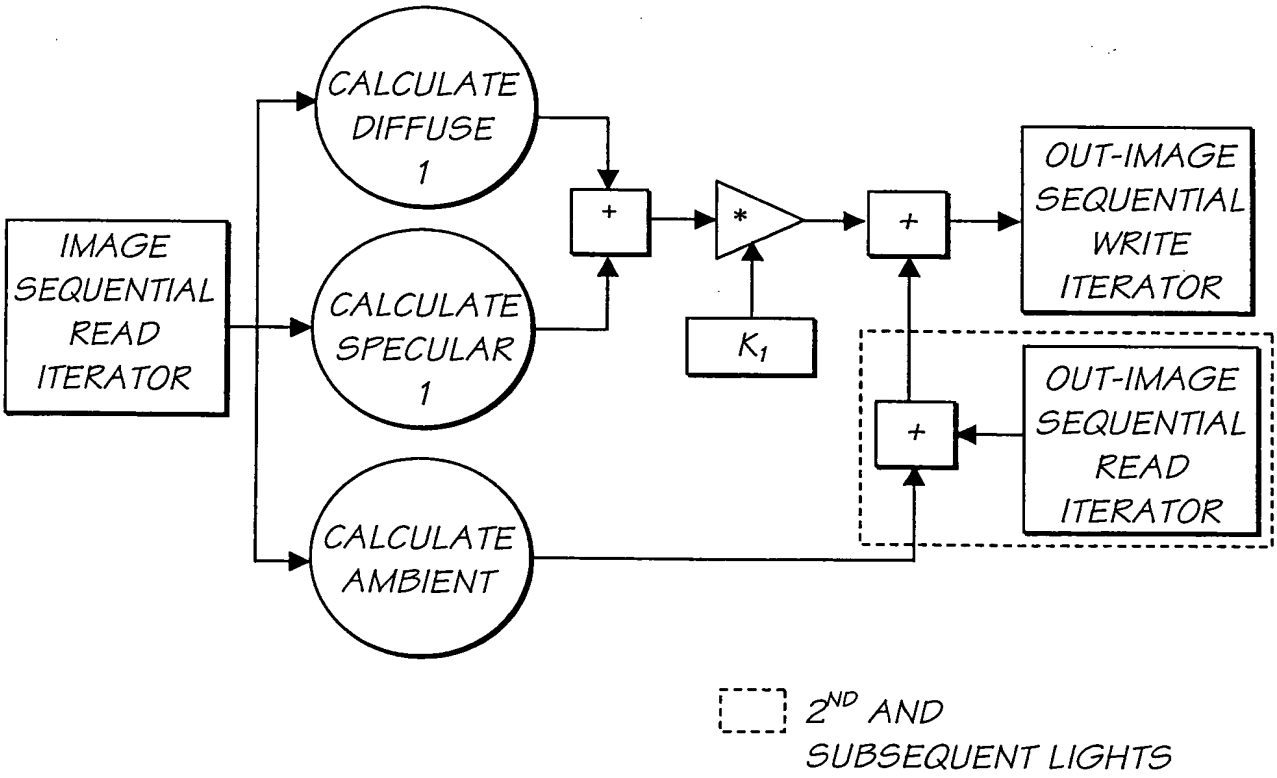


FIG. 147

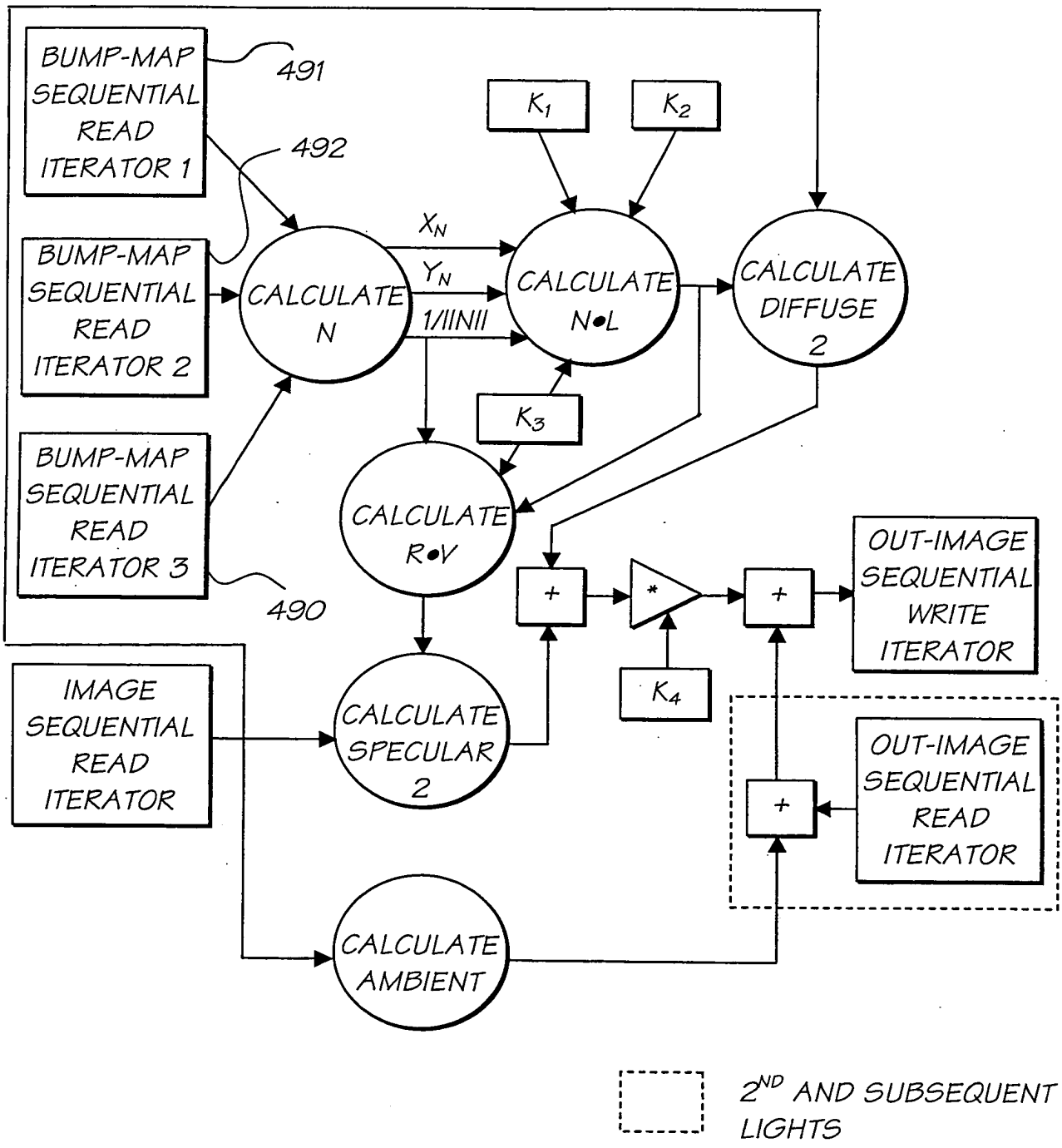

BUMP-MAP SEQUENTIAL READ ITERATOR 1
491
492
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
490
CALCULATE N
X_N
Y_N
1/||N||
K_1
K_2
CALCULATE N•L
CALCULATE DIFFUSE 2
K_3
CALCULATE R•V
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
K_4
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 148

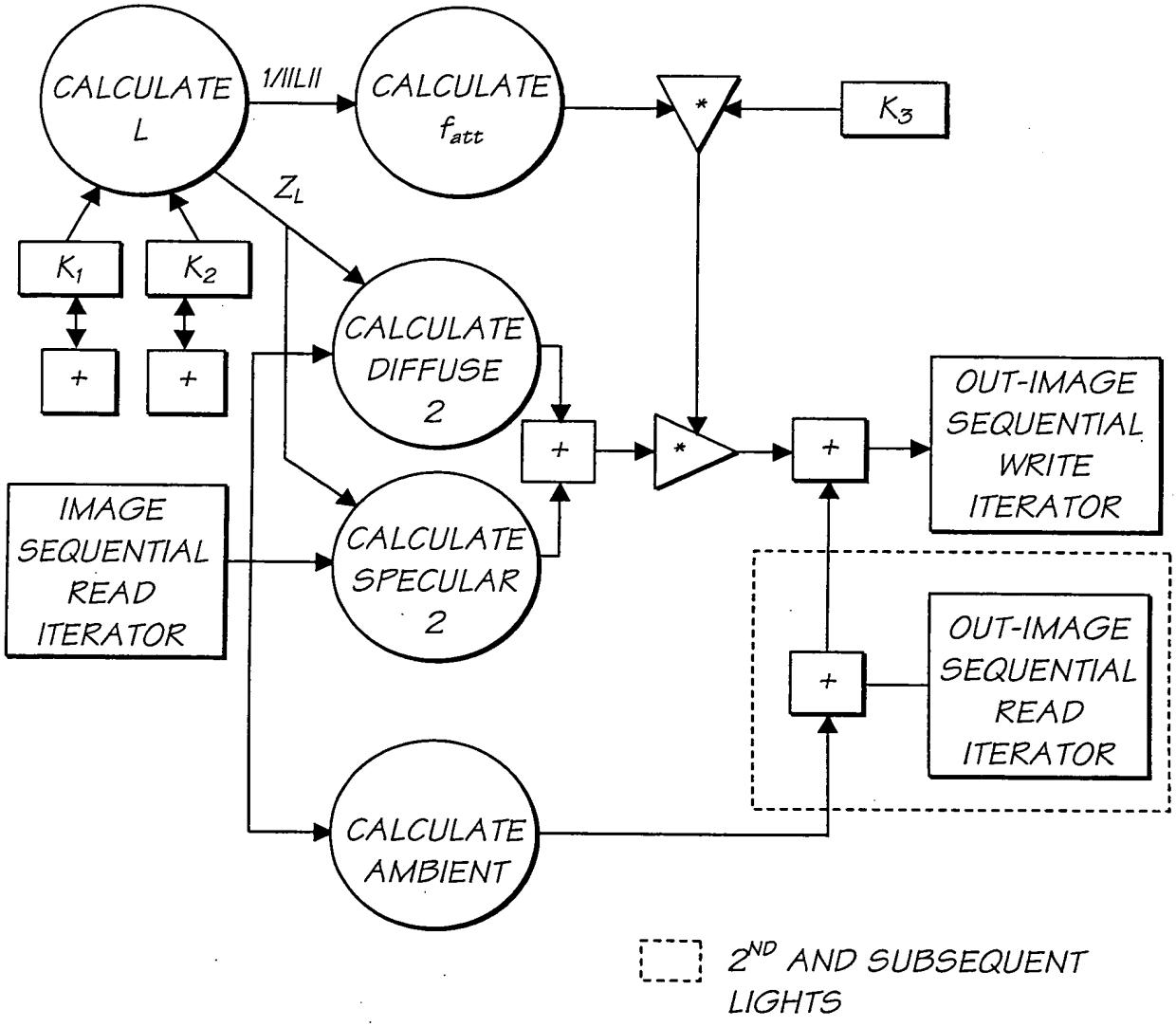

CALCULATE L
1/||L||
CALCULATE f_{att}
*
K_3
Z_L
K_1
K_2
+
+
CALCULATE DIFFUSE 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 149

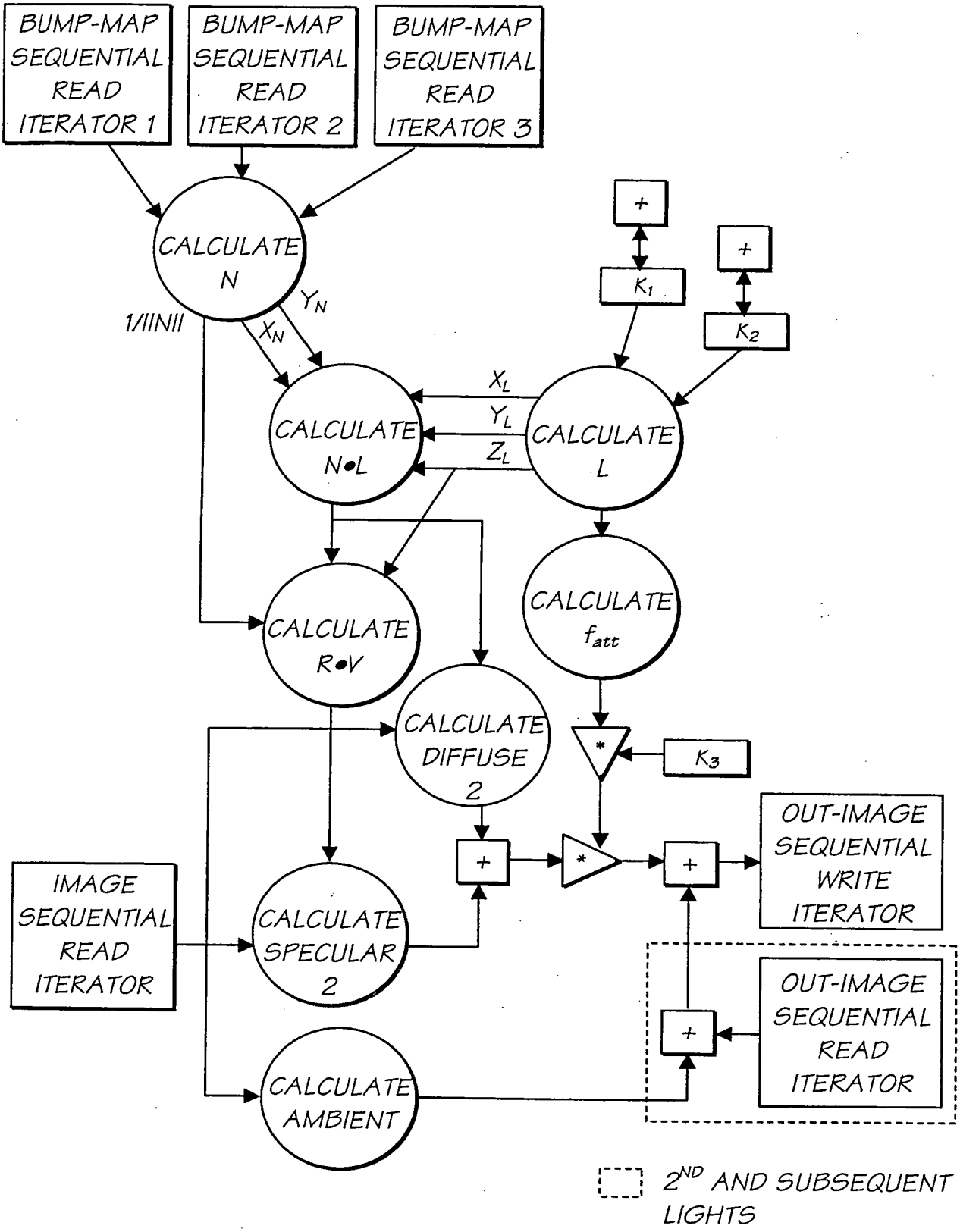

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
CALCULATE N
1/||N||
Y_N
X_N
+
K_1
+
K_2
X_L
Y_L
Z_L
CALCULATE N•L
CALCULATE L
CALCULATE R•V
CALCULATE f_{att}
CALCULATE DIFFUSE 2
*
K_3
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2^{ND} AND SUBSEQUENT LIGHTS

FIG. 150

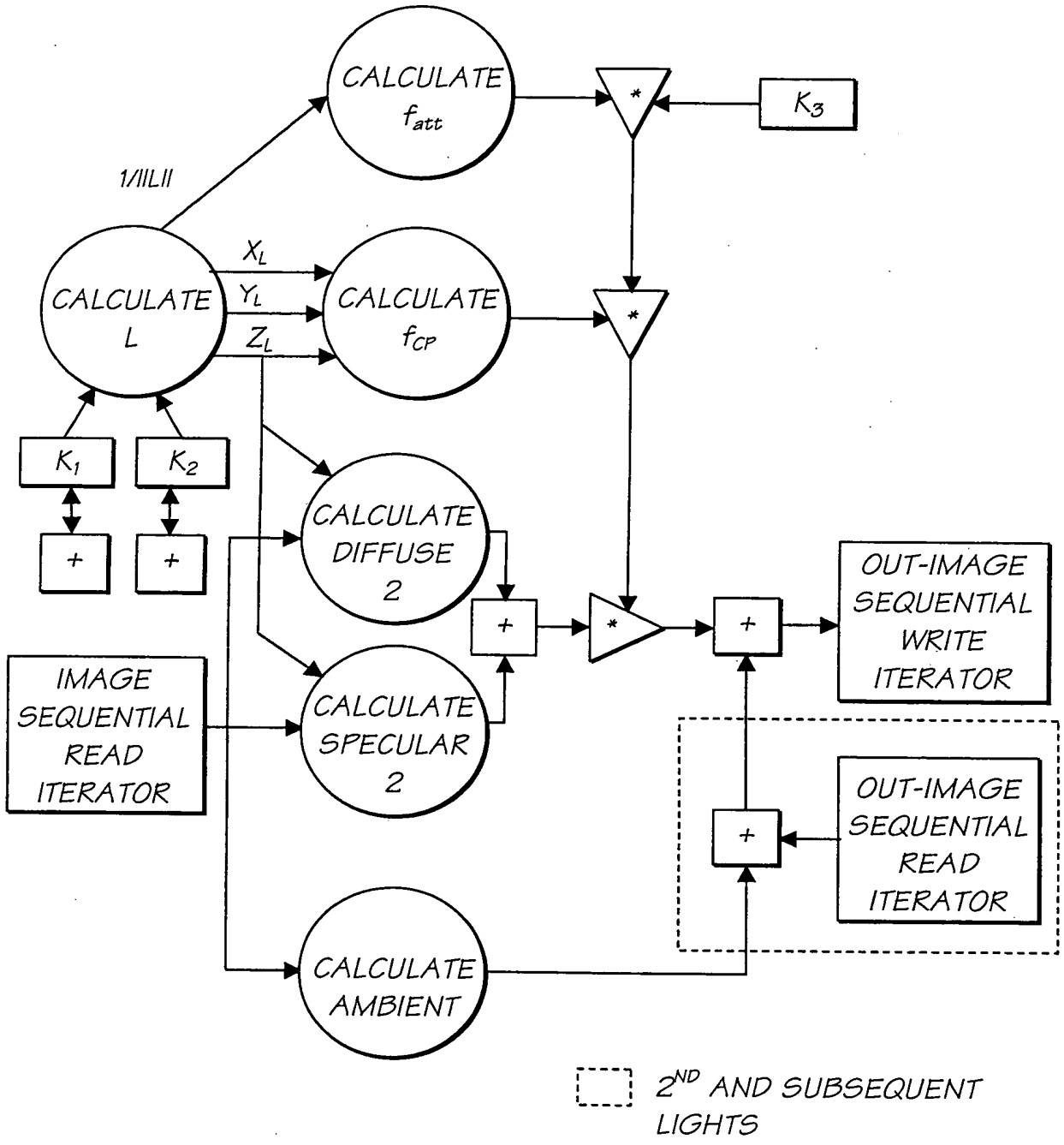

CALCULATE f_{att}
*
K_3
1/||L||
CALCULATE L
X_L
Y_L
Z_L
CALCULATE f_{CP}
*
K_1
K_2
+
+
CALCULATE DIFFUSE 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 151

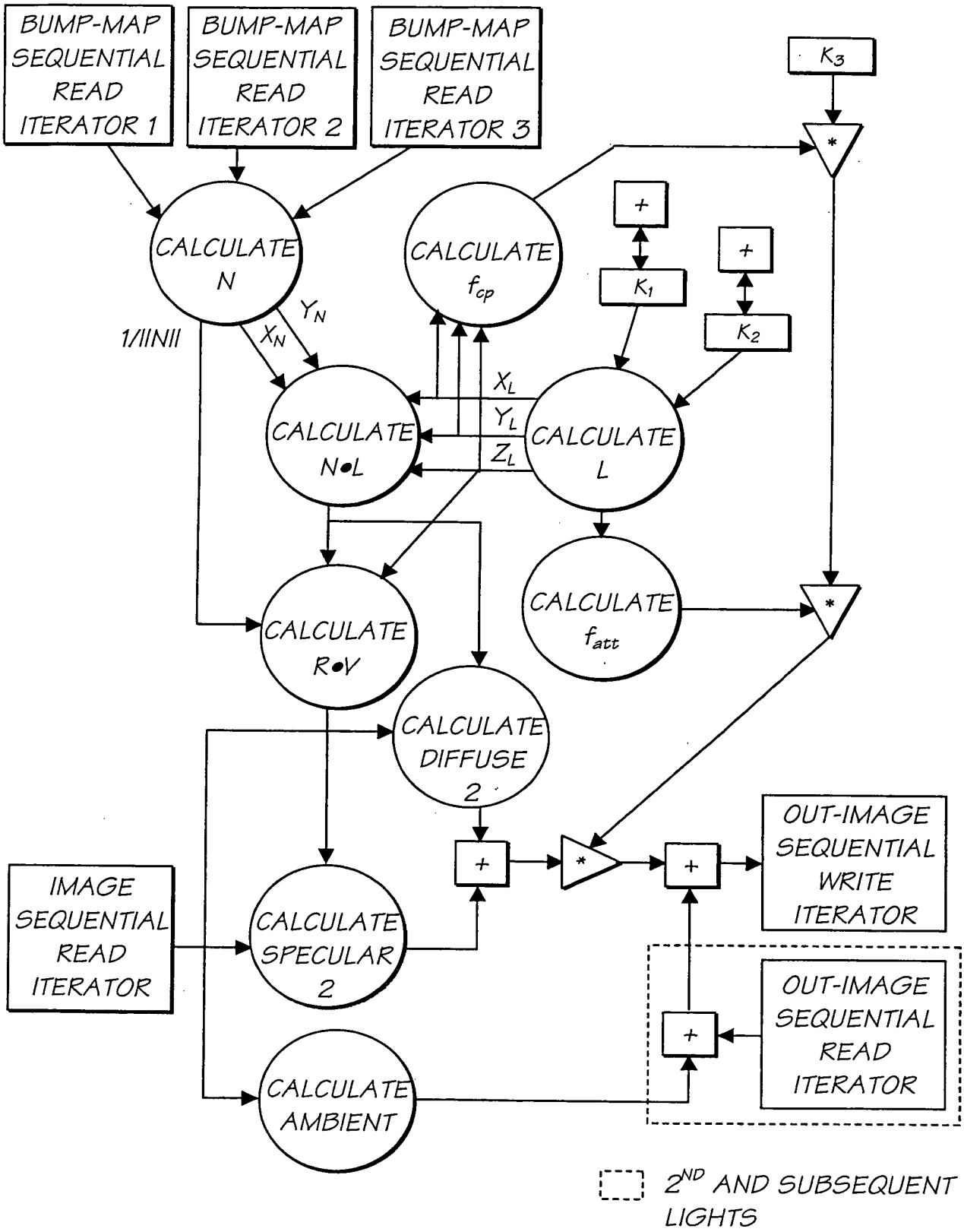
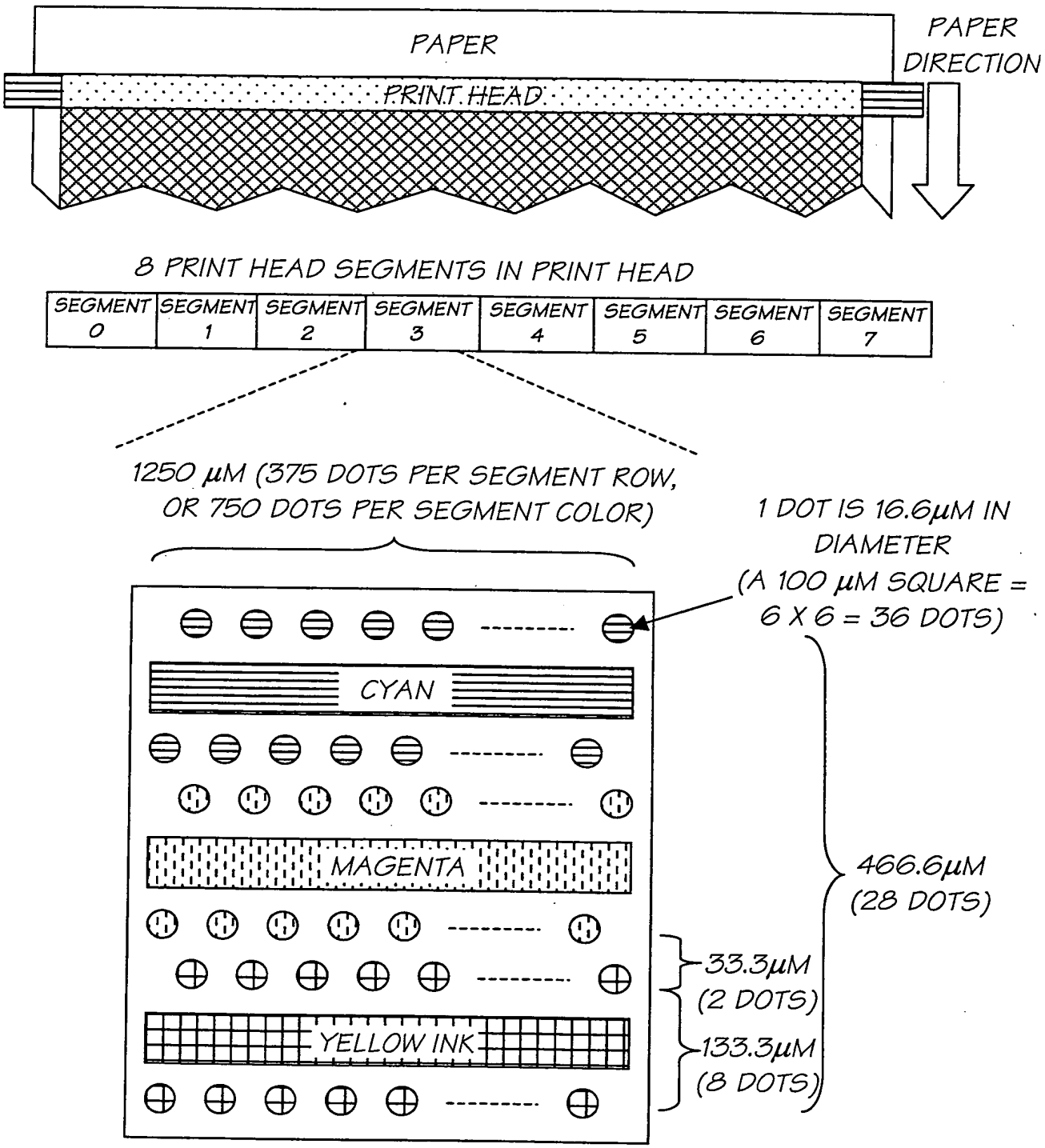

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
K3
CALCULATE N
CALCULATE f_{cp}
K1
K2
1/||N||
Y_N
X_N
X_L
Y_L
Z_L
CALCULATE N•L
CALCULATE L
CALCULATE R•V
CALCULATE f_{att}
CALCULATE DIFFUSE 2
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 152



EACH SEGMENT CONTAINS 6 ROWS OF DOTS: ODD AND EVEN CYAN, MAGENTA, AND YELLOW.

FIG. 153

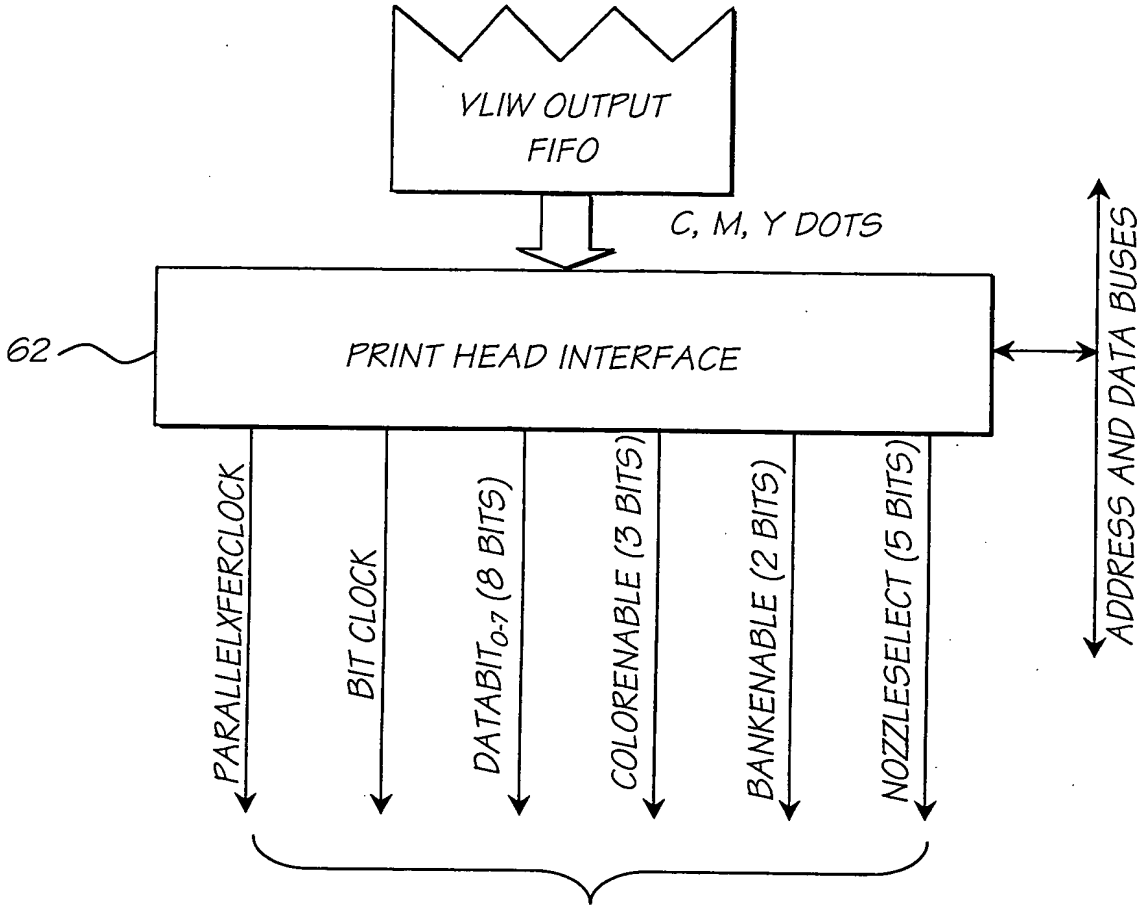


FIG. 154

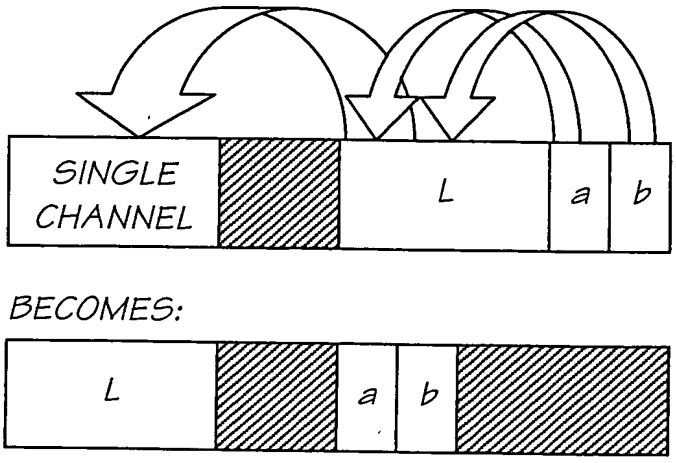


FIG. 155

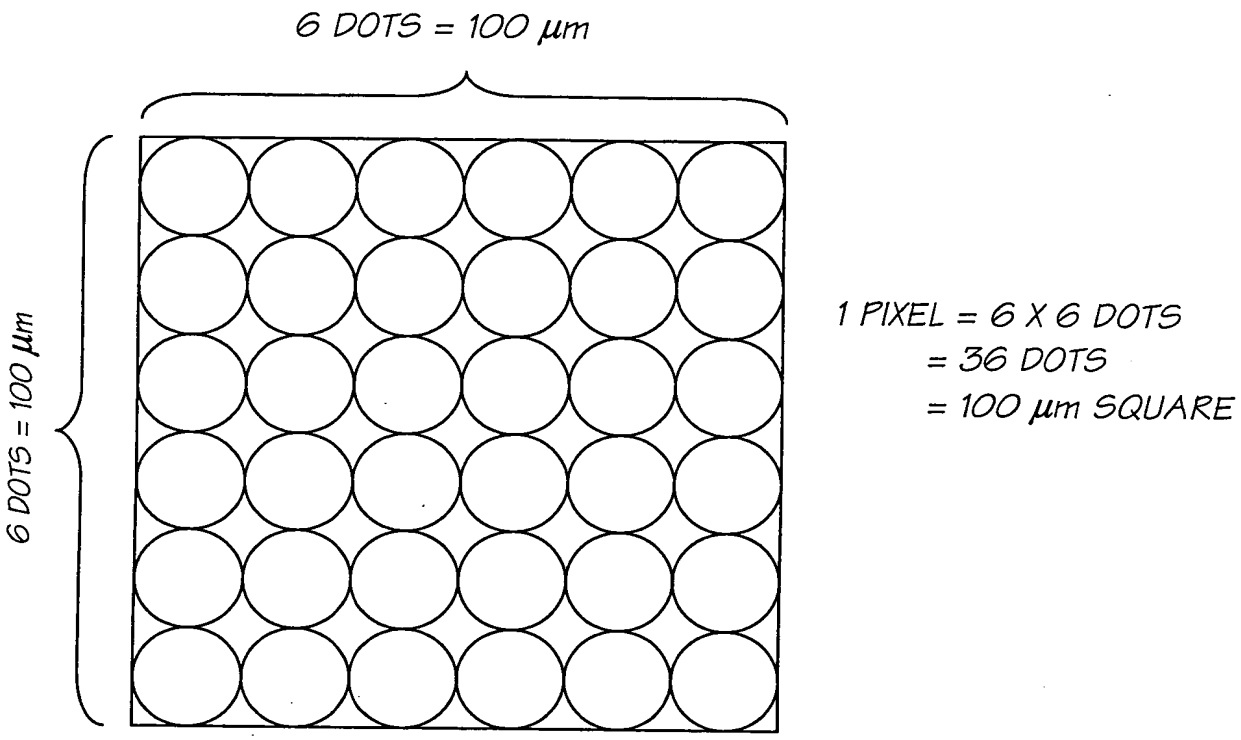

6 DOTS = 100 μm
6 DOTS = 100 μm
1 PIXEL = 6 X 6 DOTS
= 36 DOTS
= 100 μm SQUARE

FIG. 156

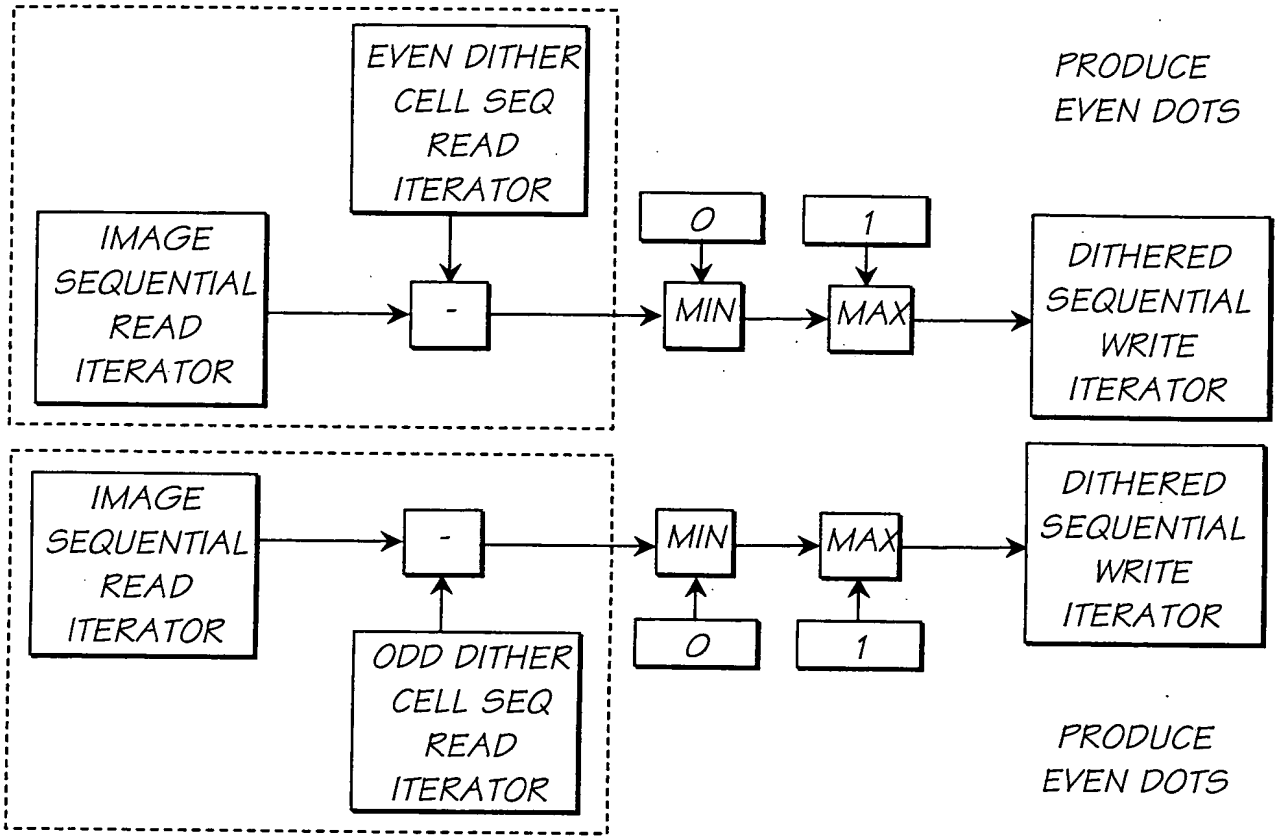


FIG. 157

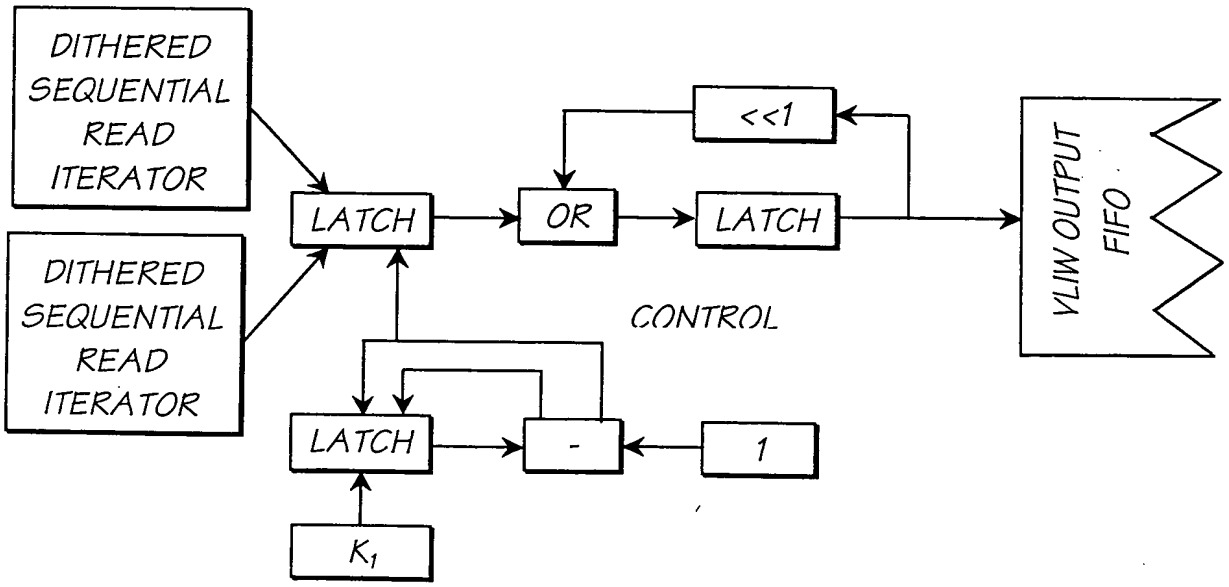


FIG. 158

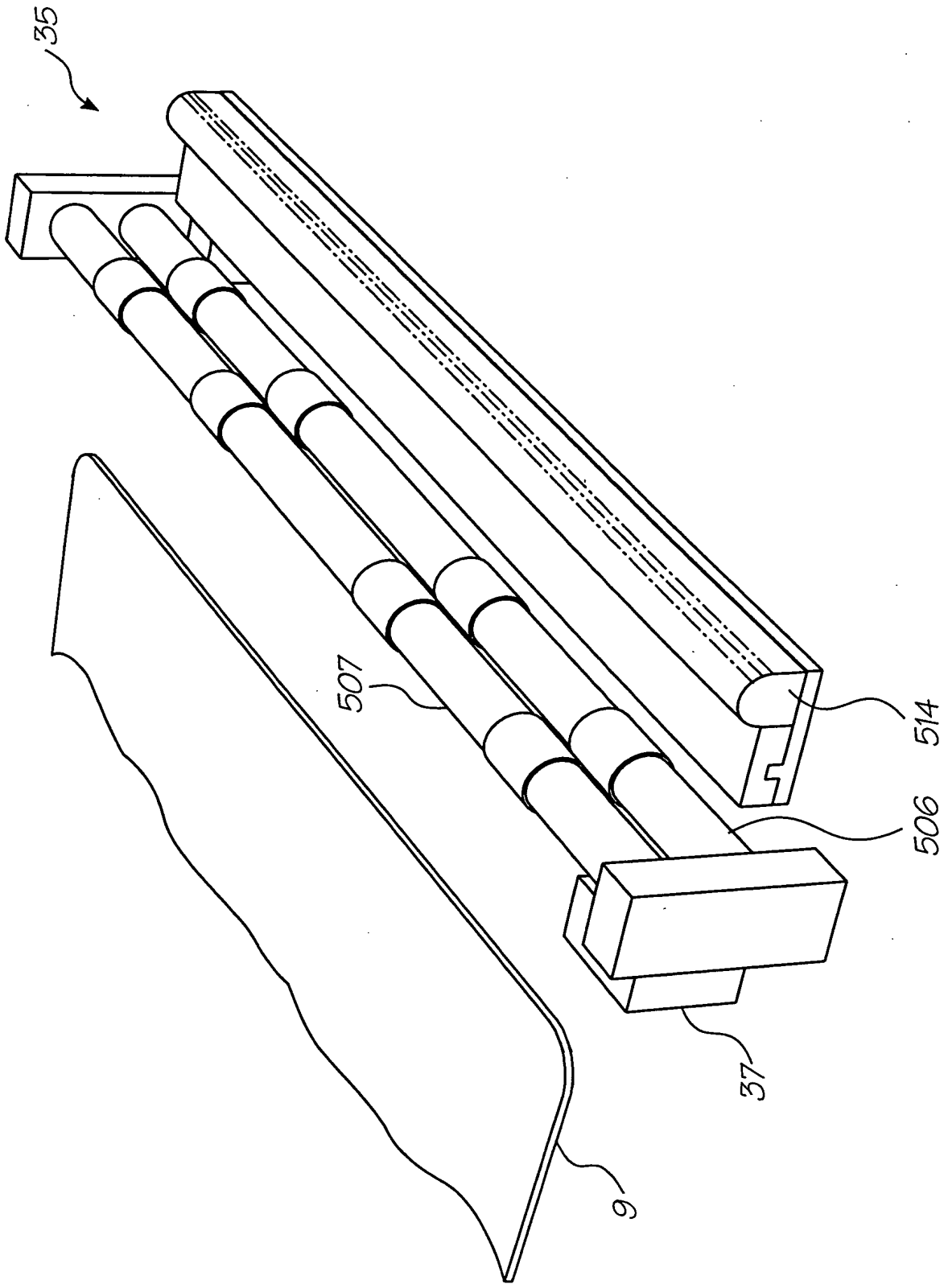


FIG. 159

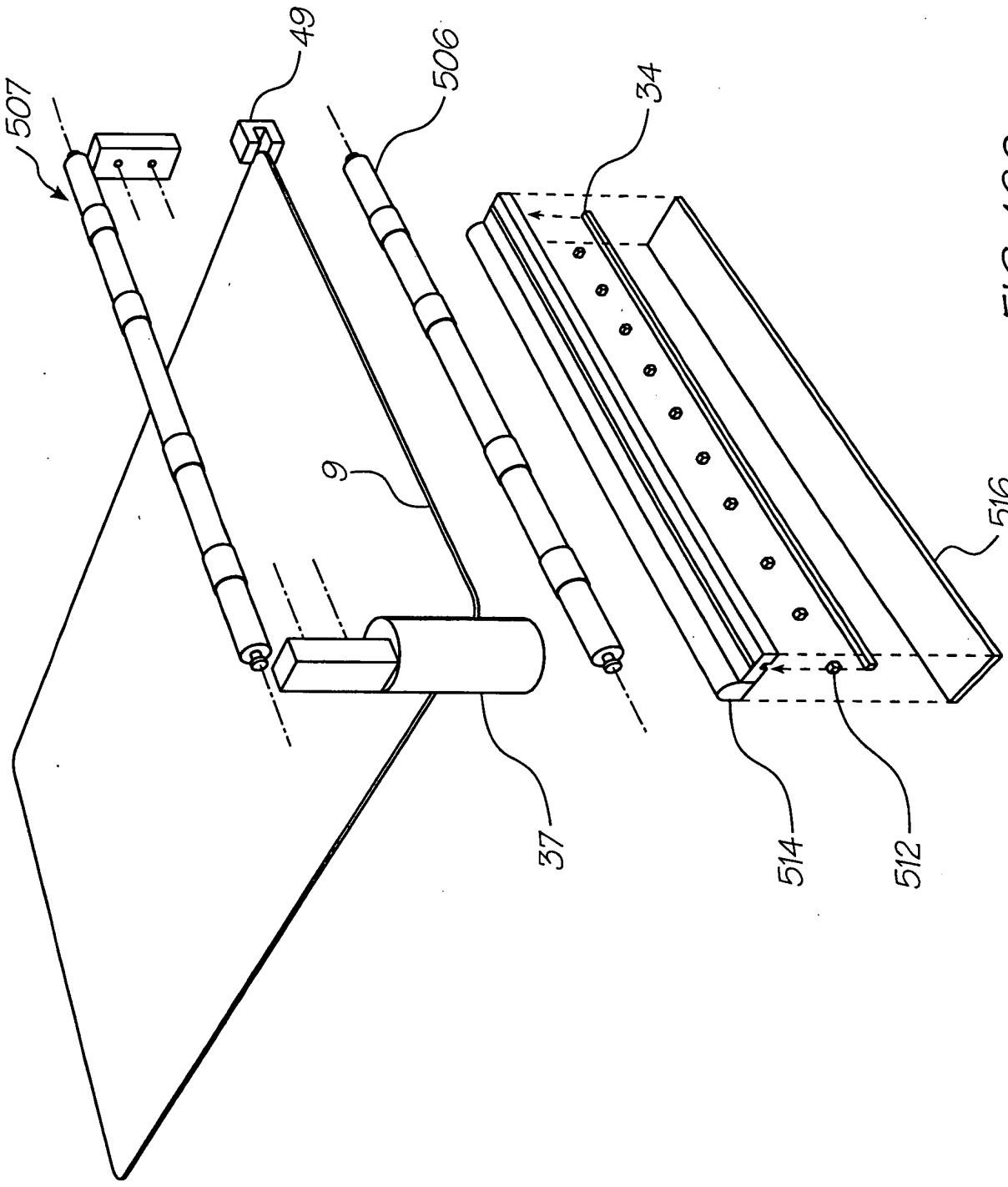


FIG. 160

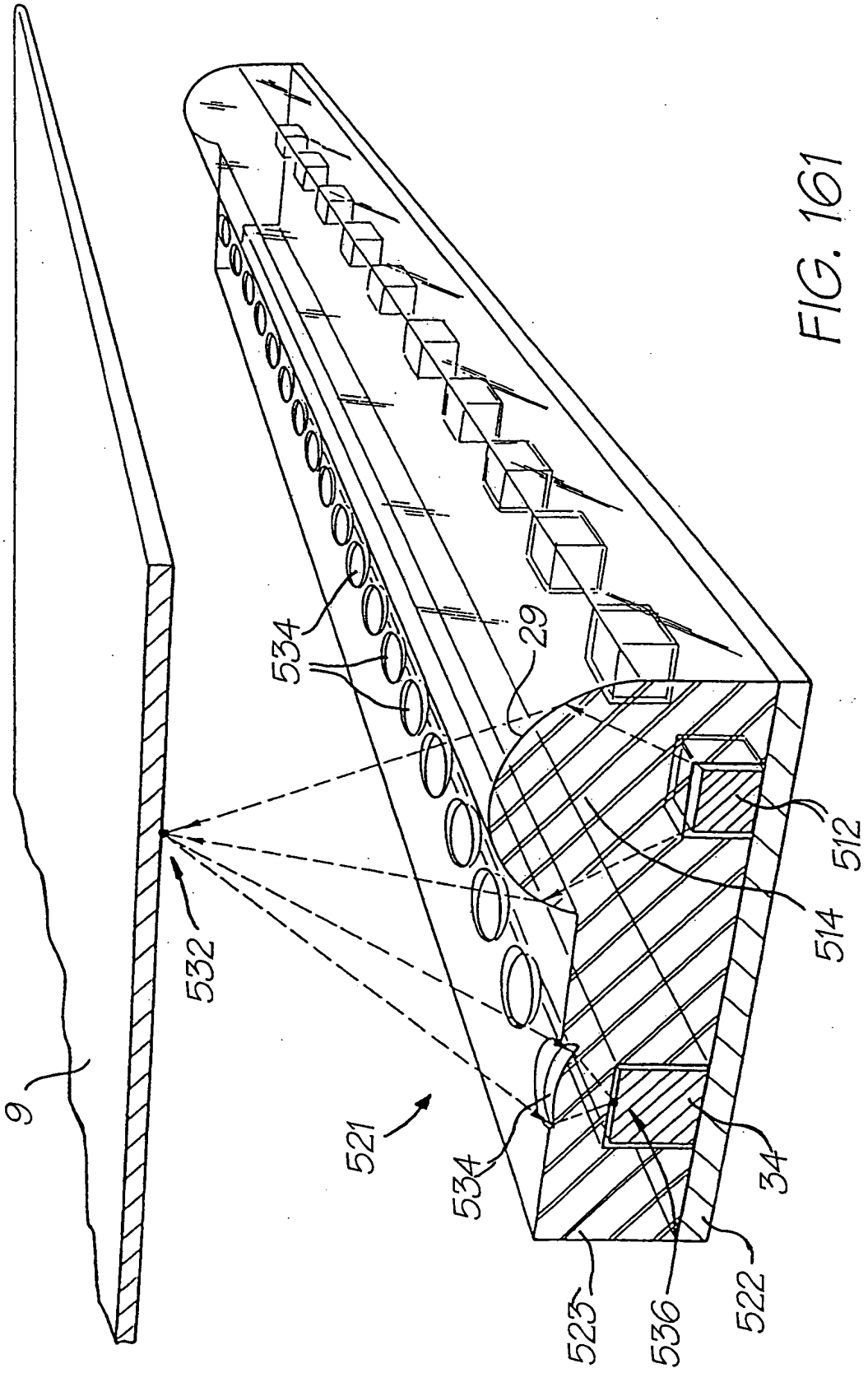


FIG. 161

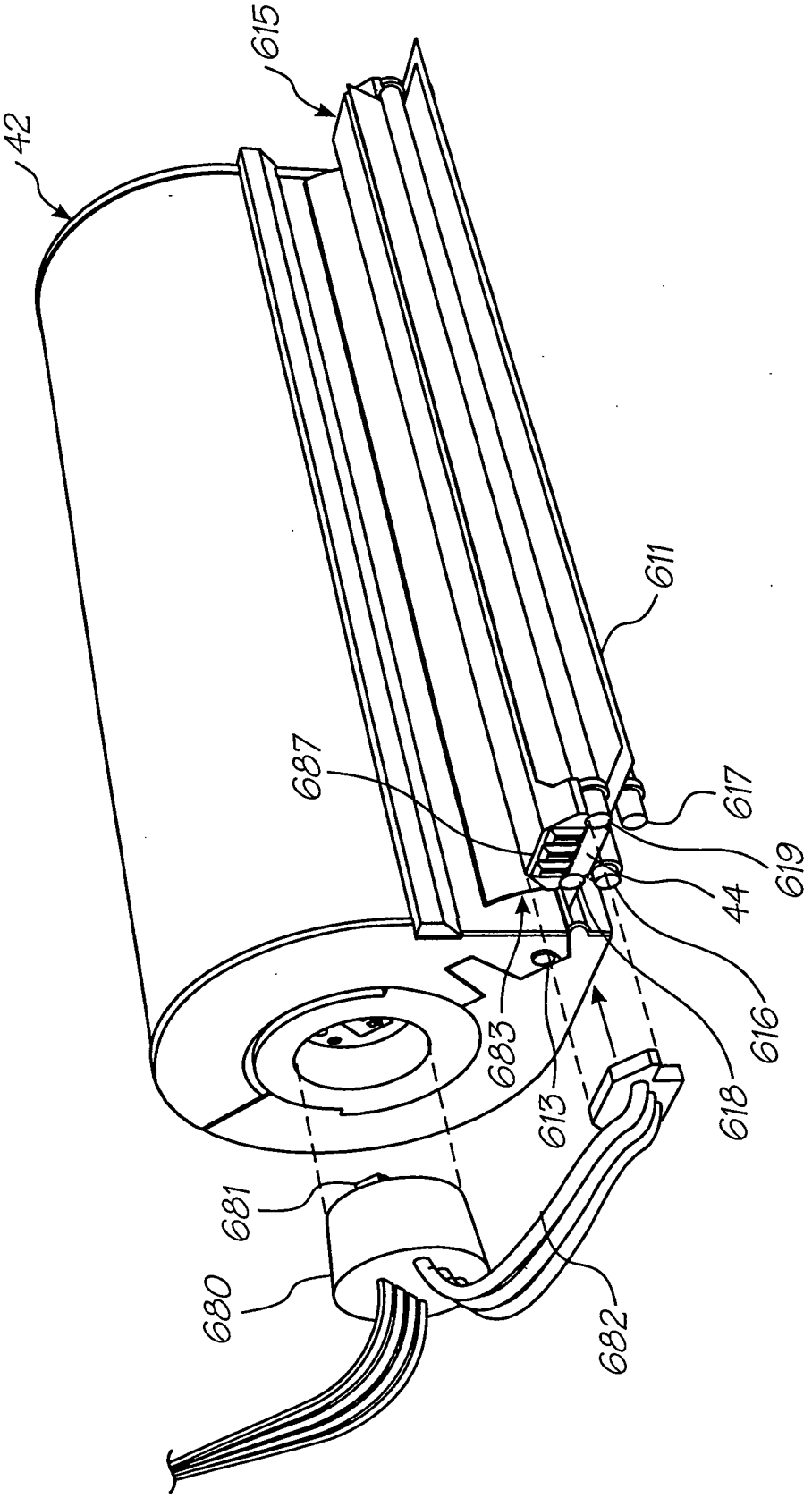


FIG. 162

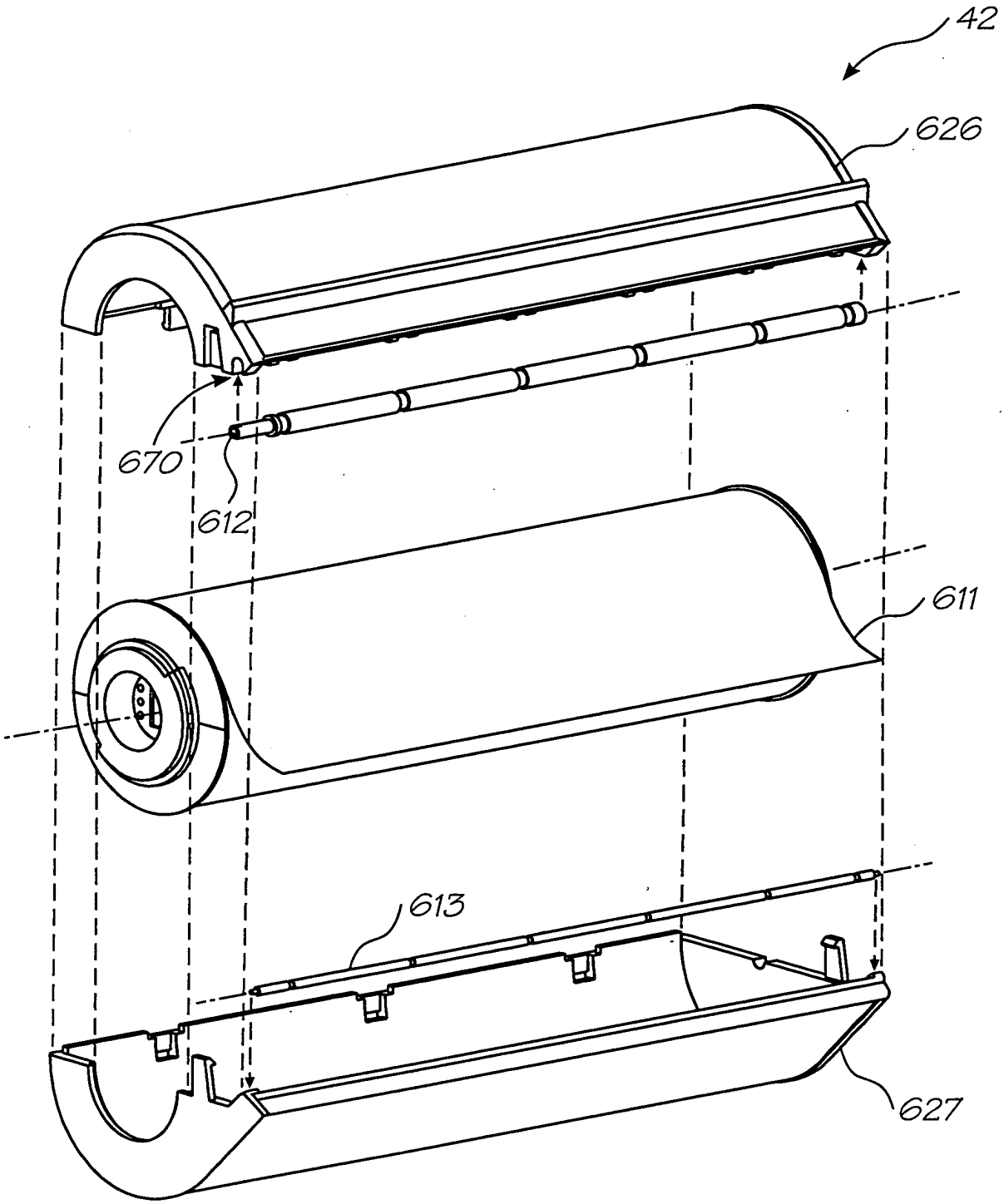


FIG. 163

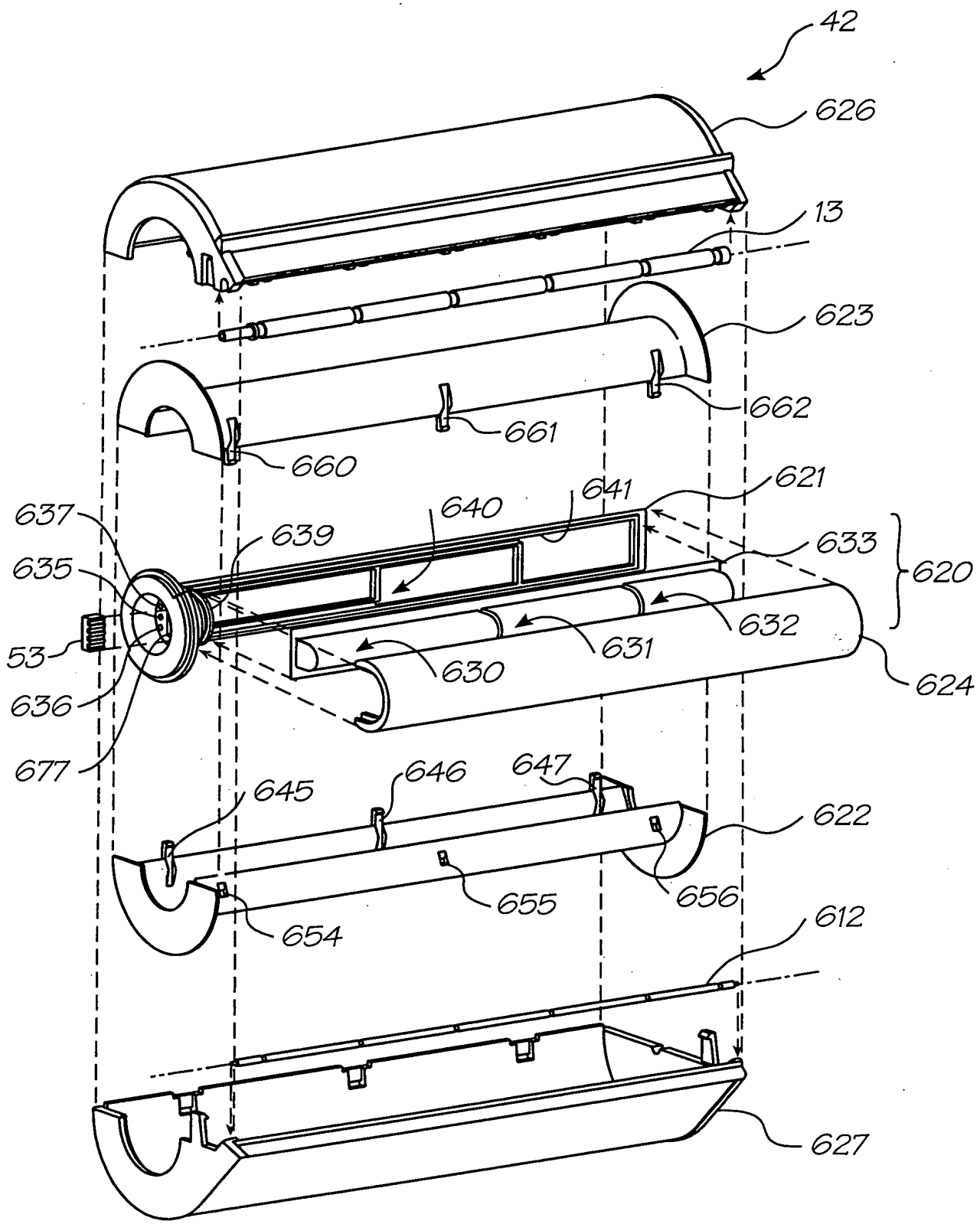

42
626
13
623
662
661
660
621
641
640
637
639
633
620
635
632
631
53
630
624
636
677
645
646
647
622
655
656
654
612
627

FIG. 164

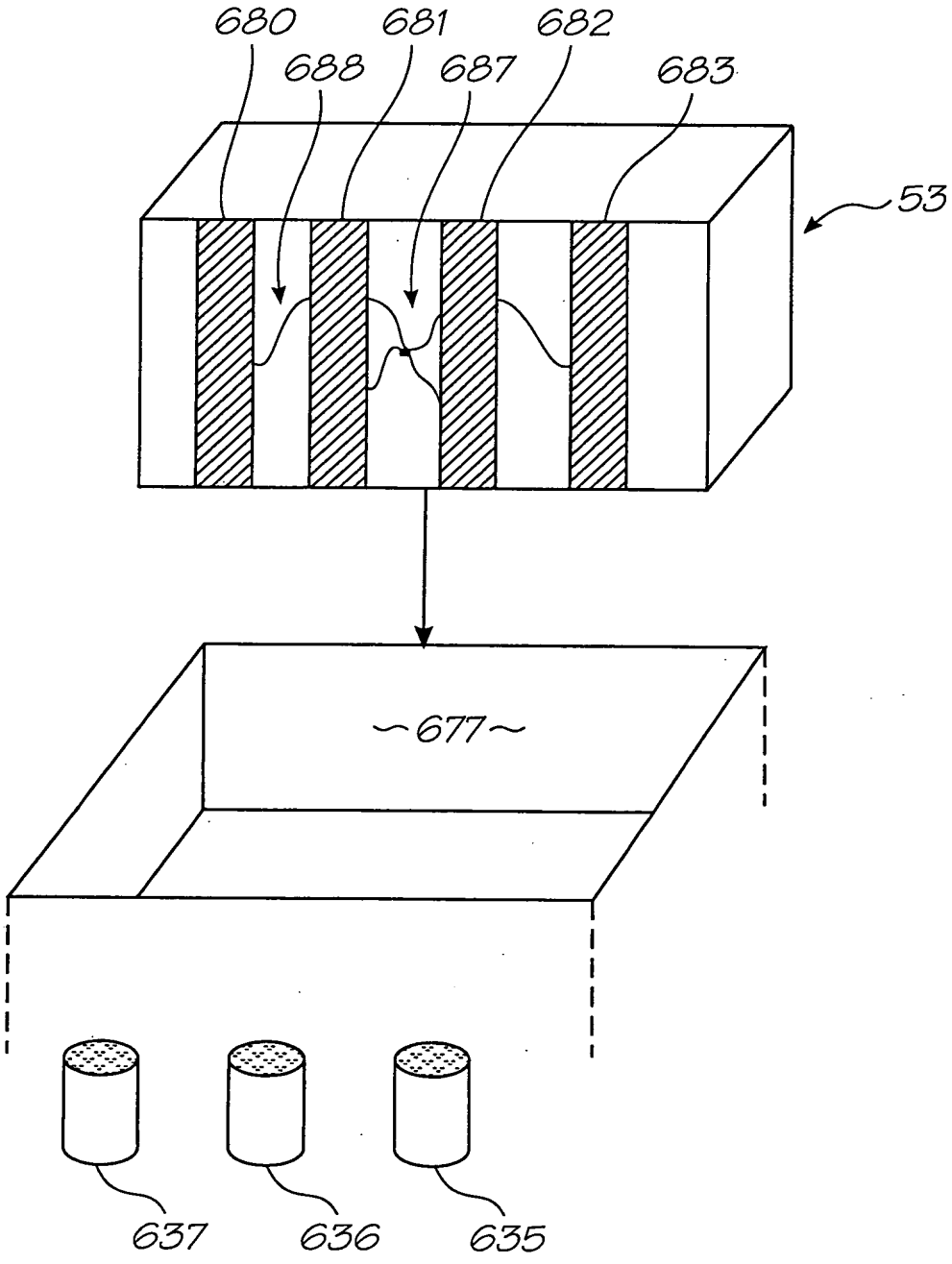


FIG. 165

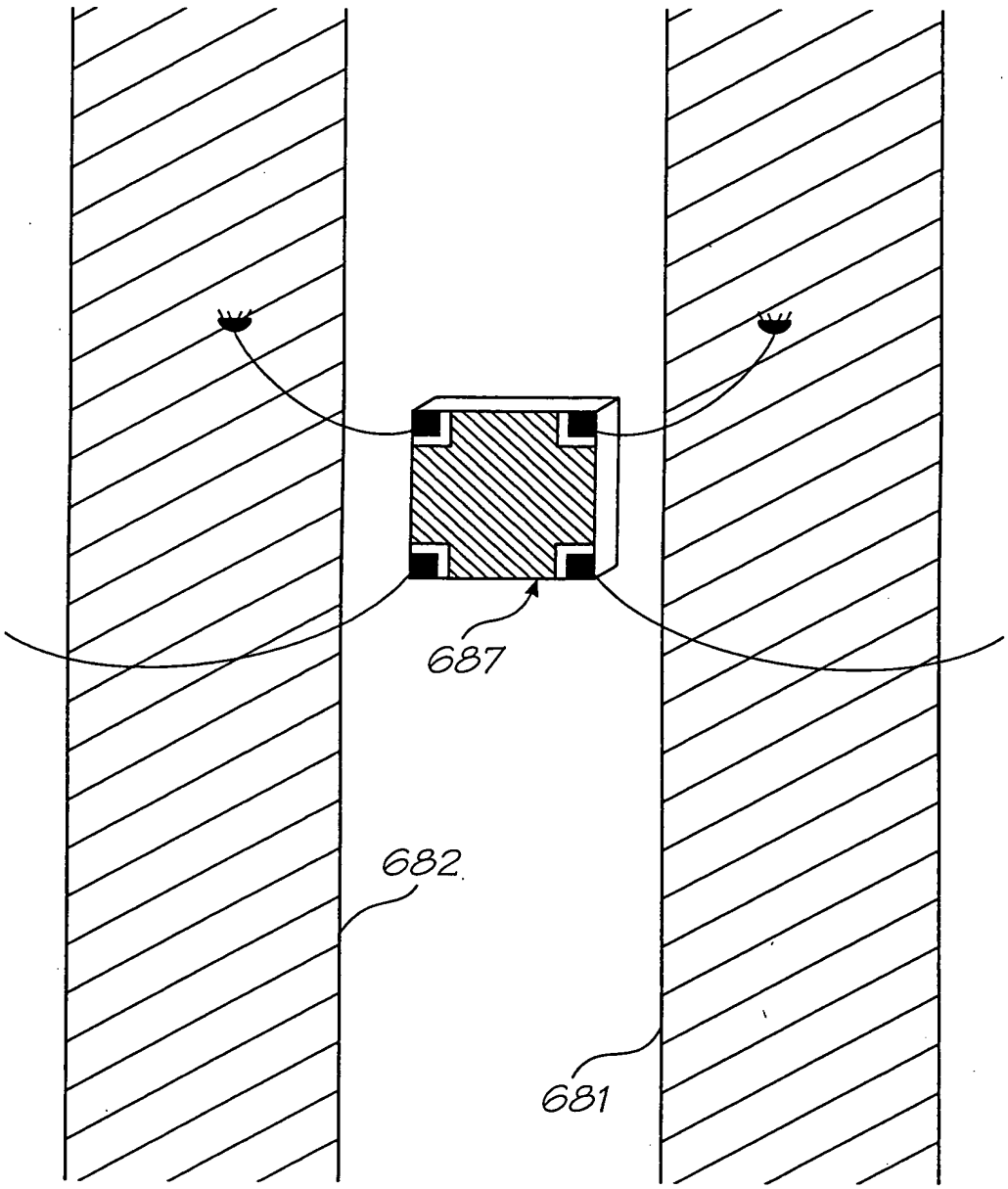


FIG. 166

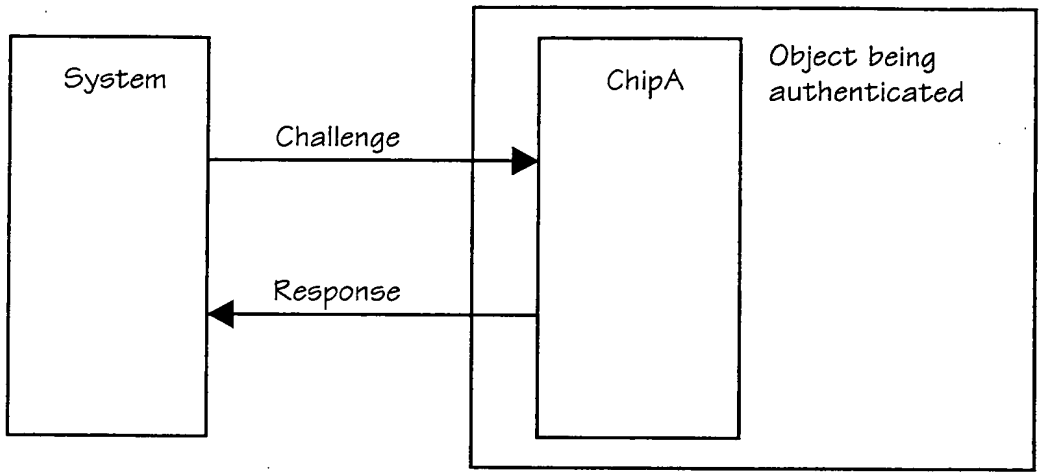


FIG. 167

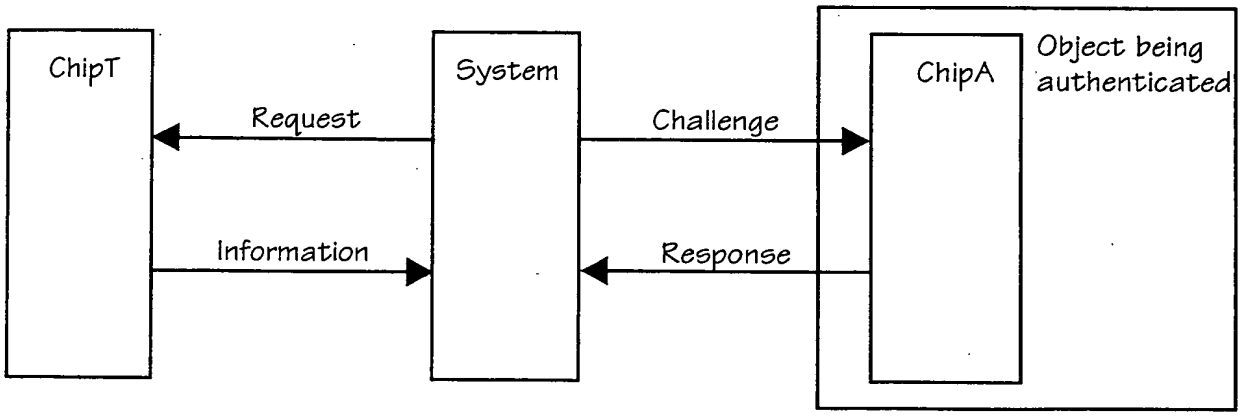


FIG. 168

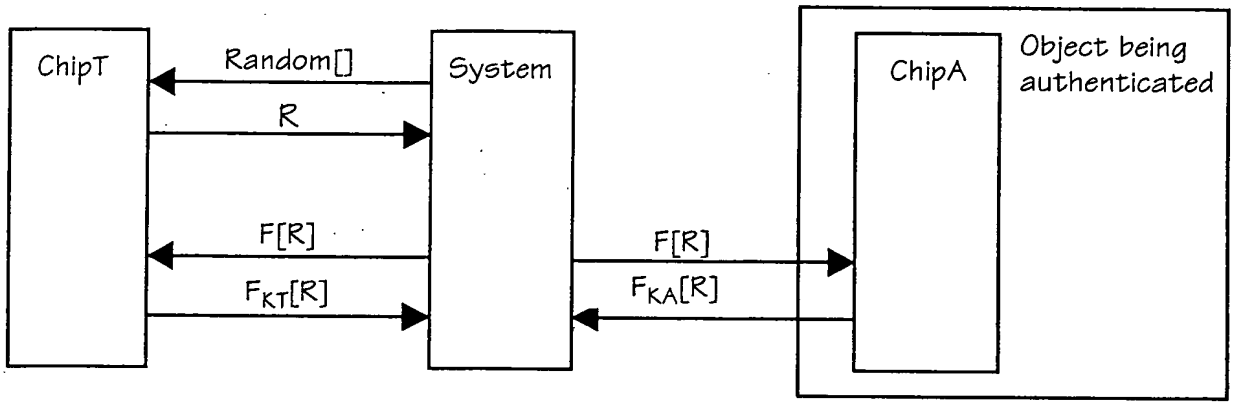


FIG. 169

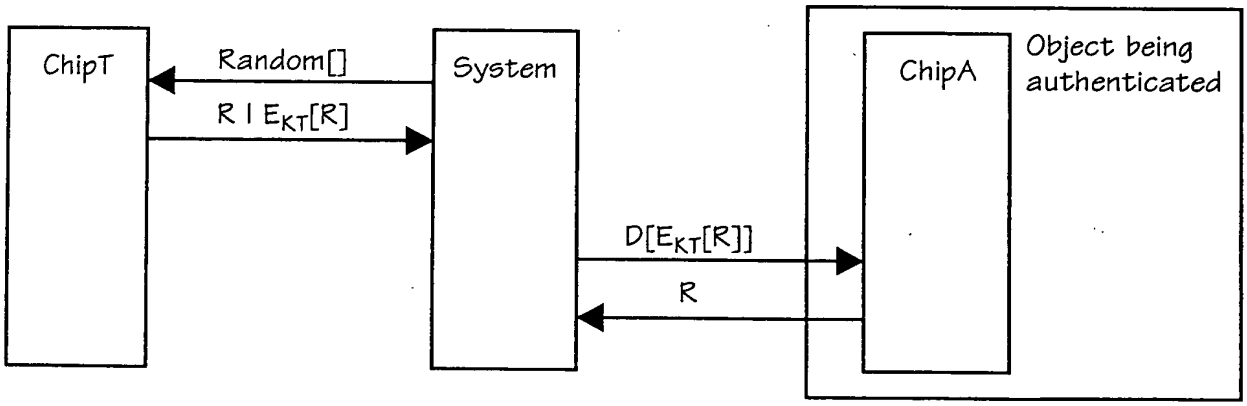


FIG. 170

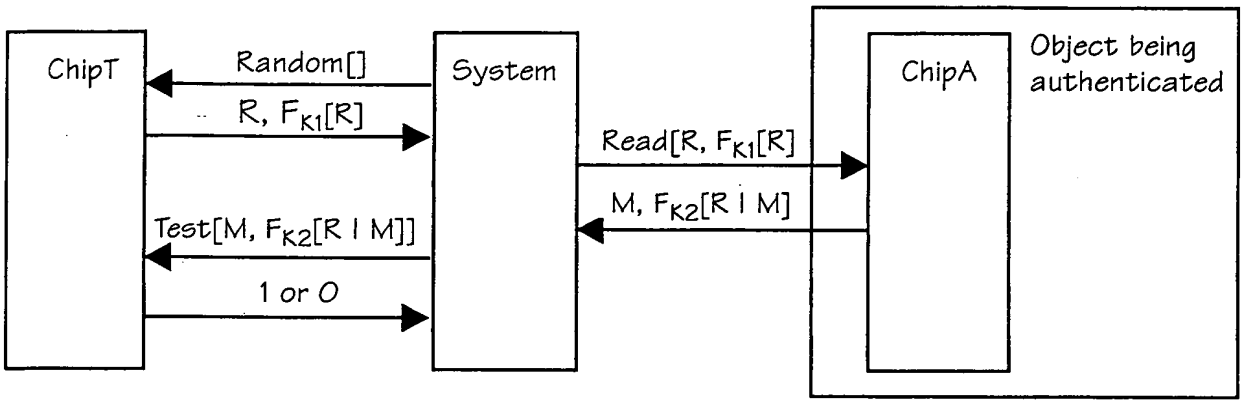


FIG. 171

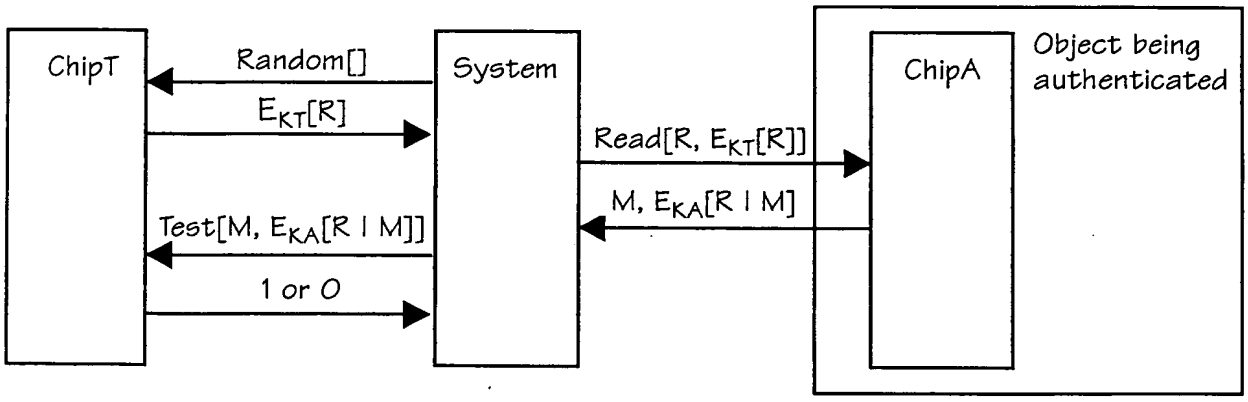


FIG. 172

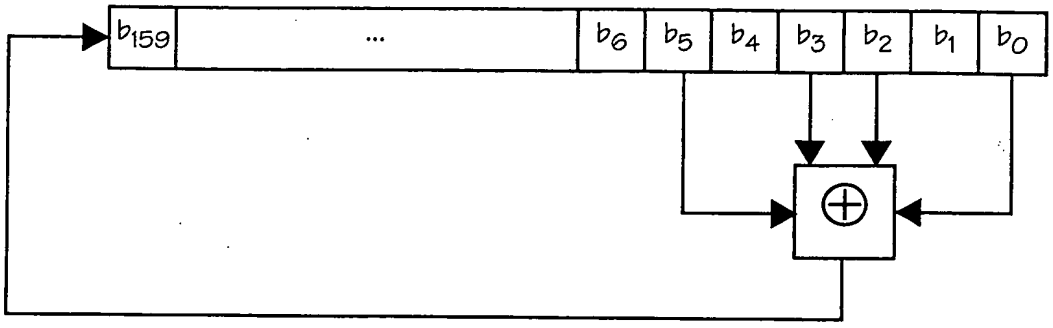


FIG. 173

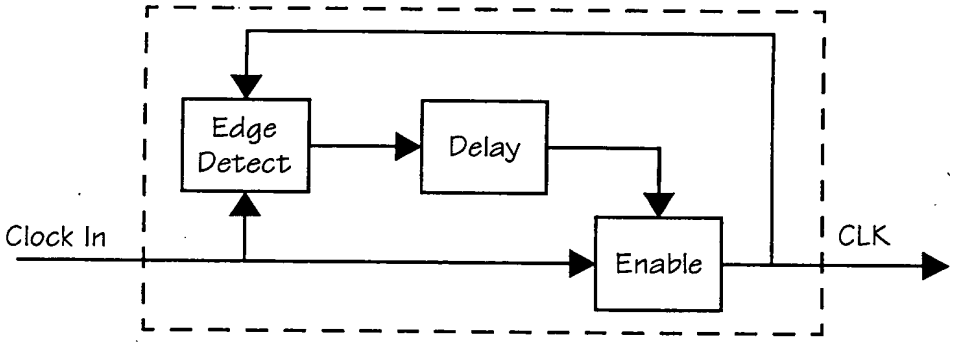


FIG. 174

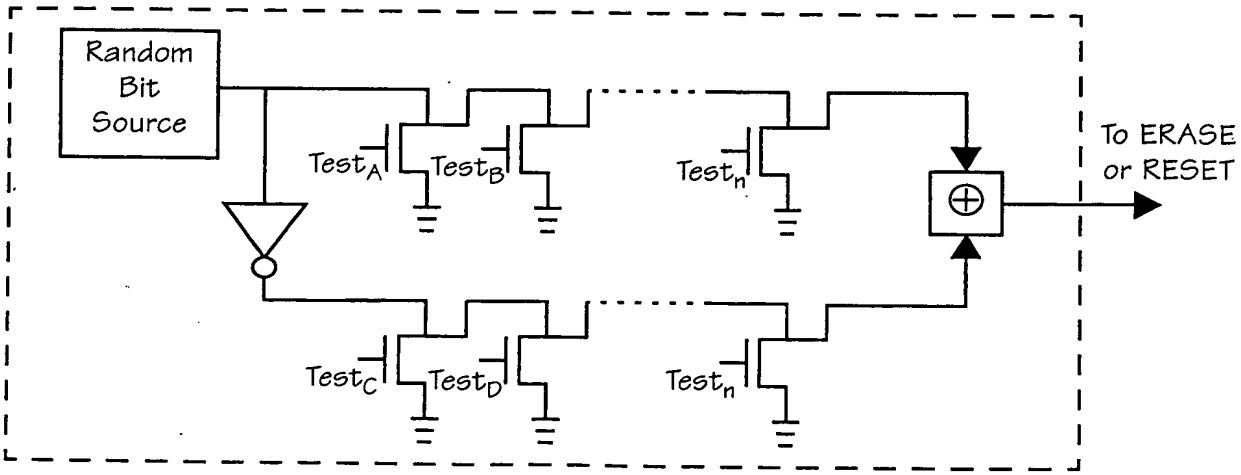


FIG. 175

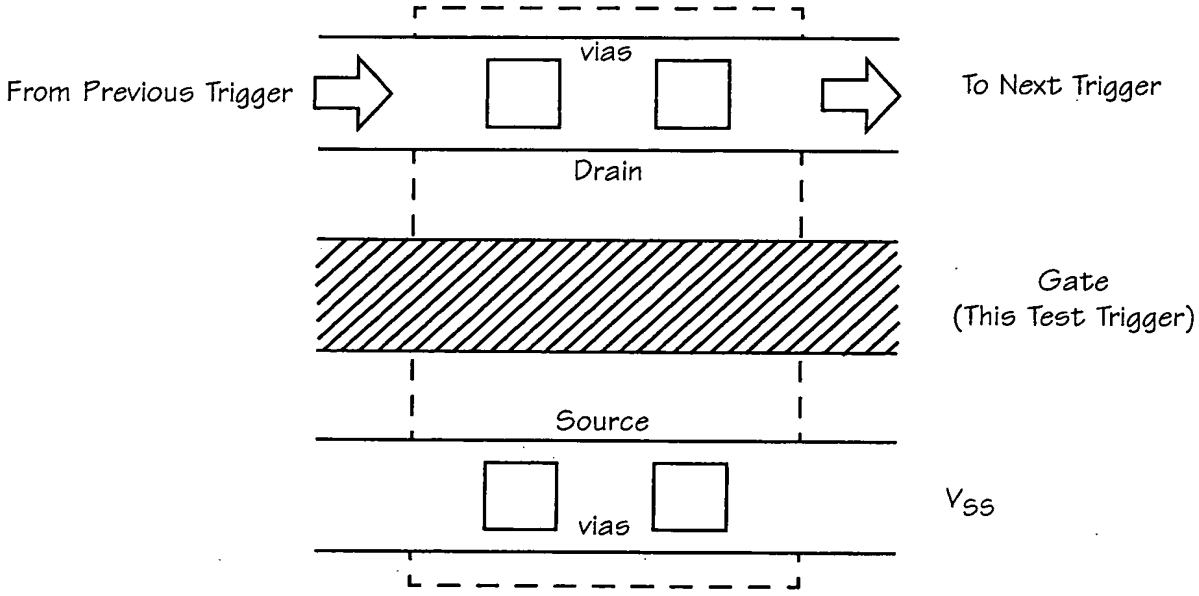


FIG. 176

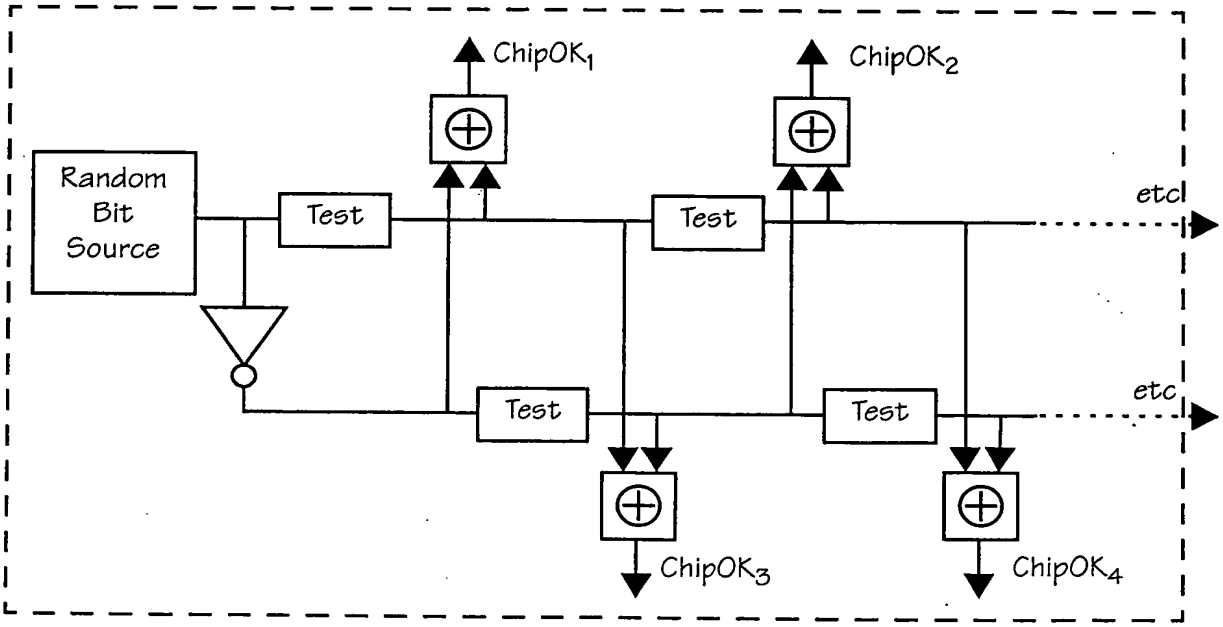


FIG. 177

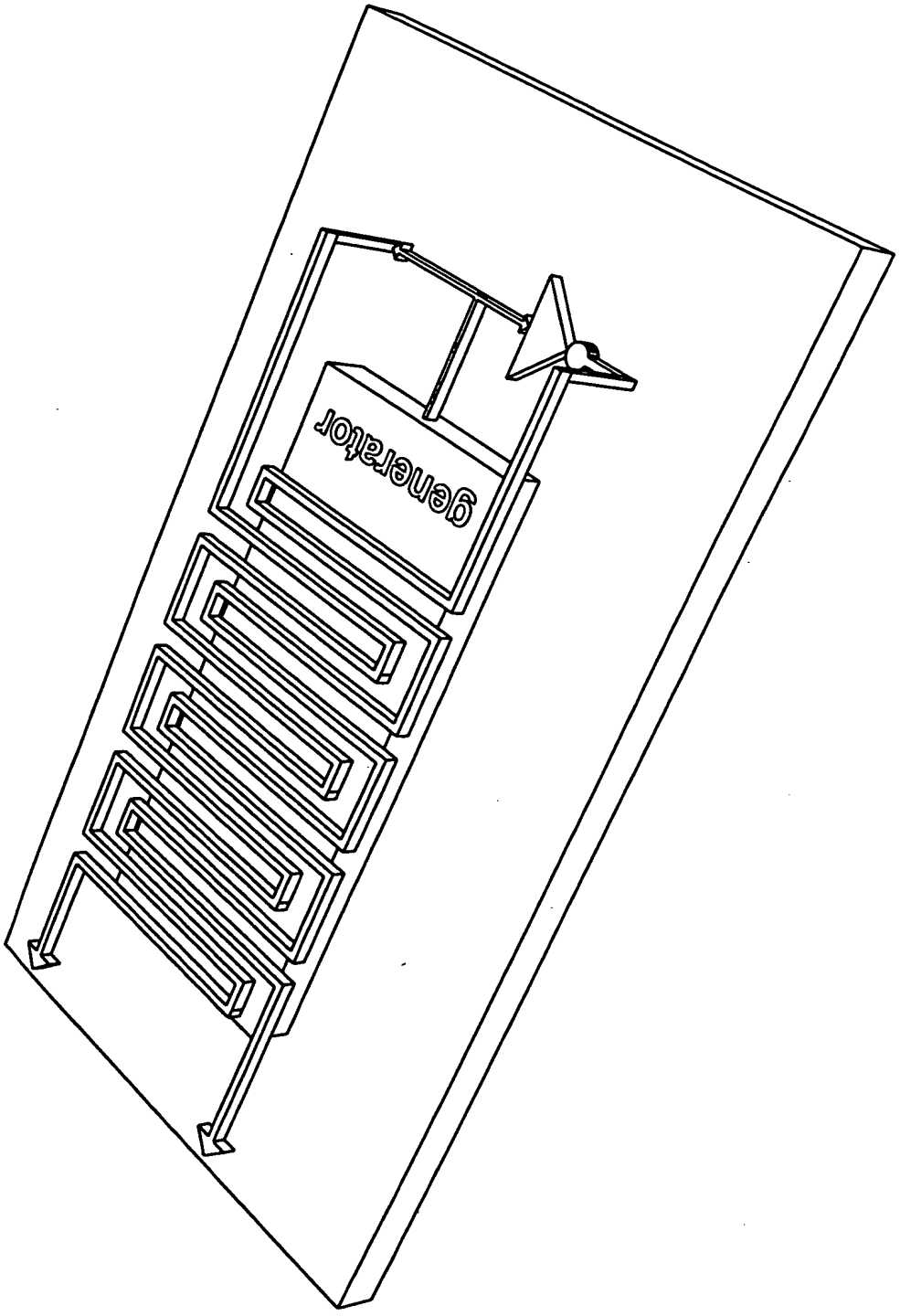


FIG. 178

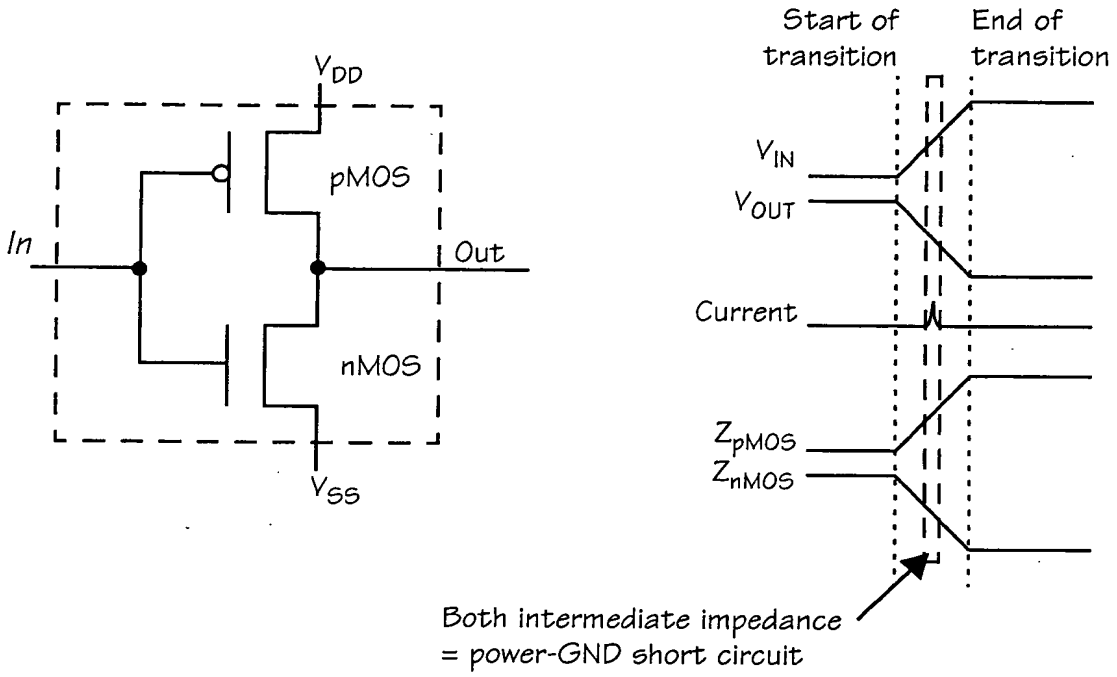


FIG. 179

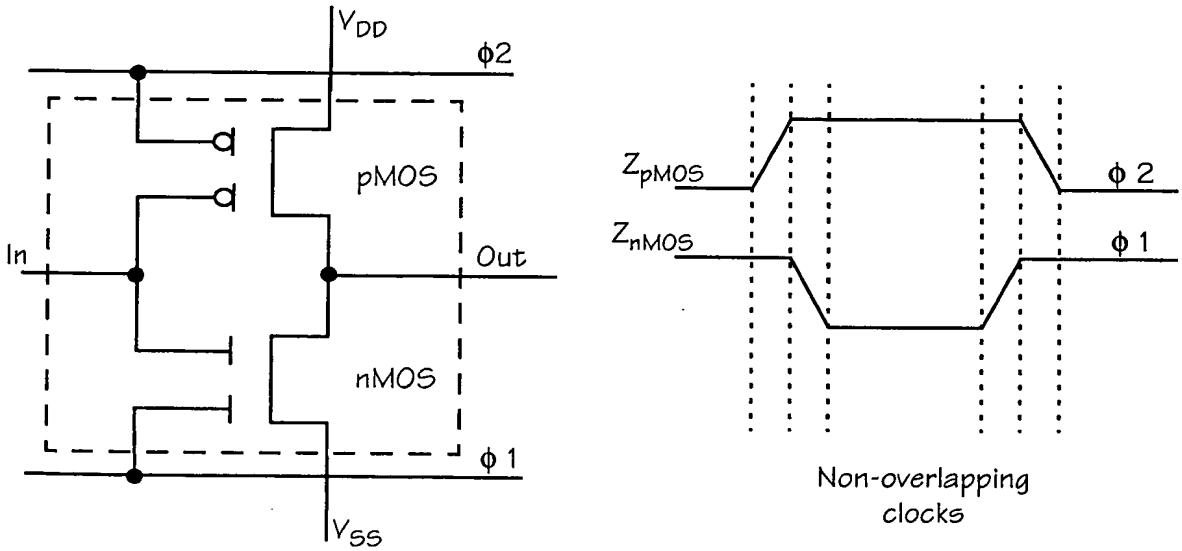


FIG. 180

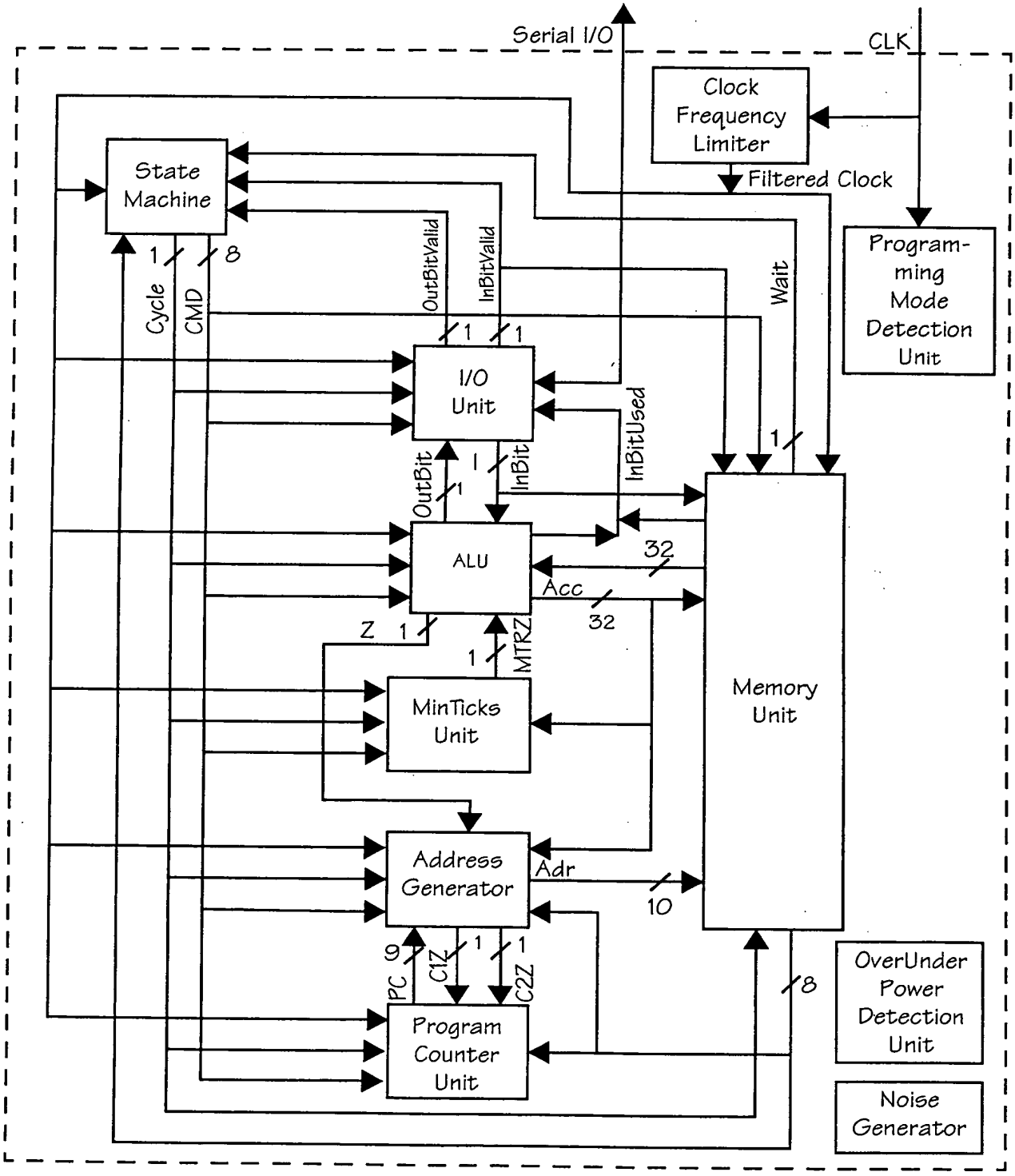

Serial I/O
CLK
Clock Frequency Limiter
Filtered Clock
State Machine
Programming Mode Detection Unit
Cycle
CMD
OutBitValid
InBitValid
Wait
I/O Unit
OutBit
InBit
InBitUsed
ALU
Acc
32
Z
MTRZ
MinTicks Unit
Memory Unit
Address Generator
Adr
10
PC
C1Z
C2Z
Program Counter Unit
OverUnder Power Detection Unit
Noise Generator

FIG. 181

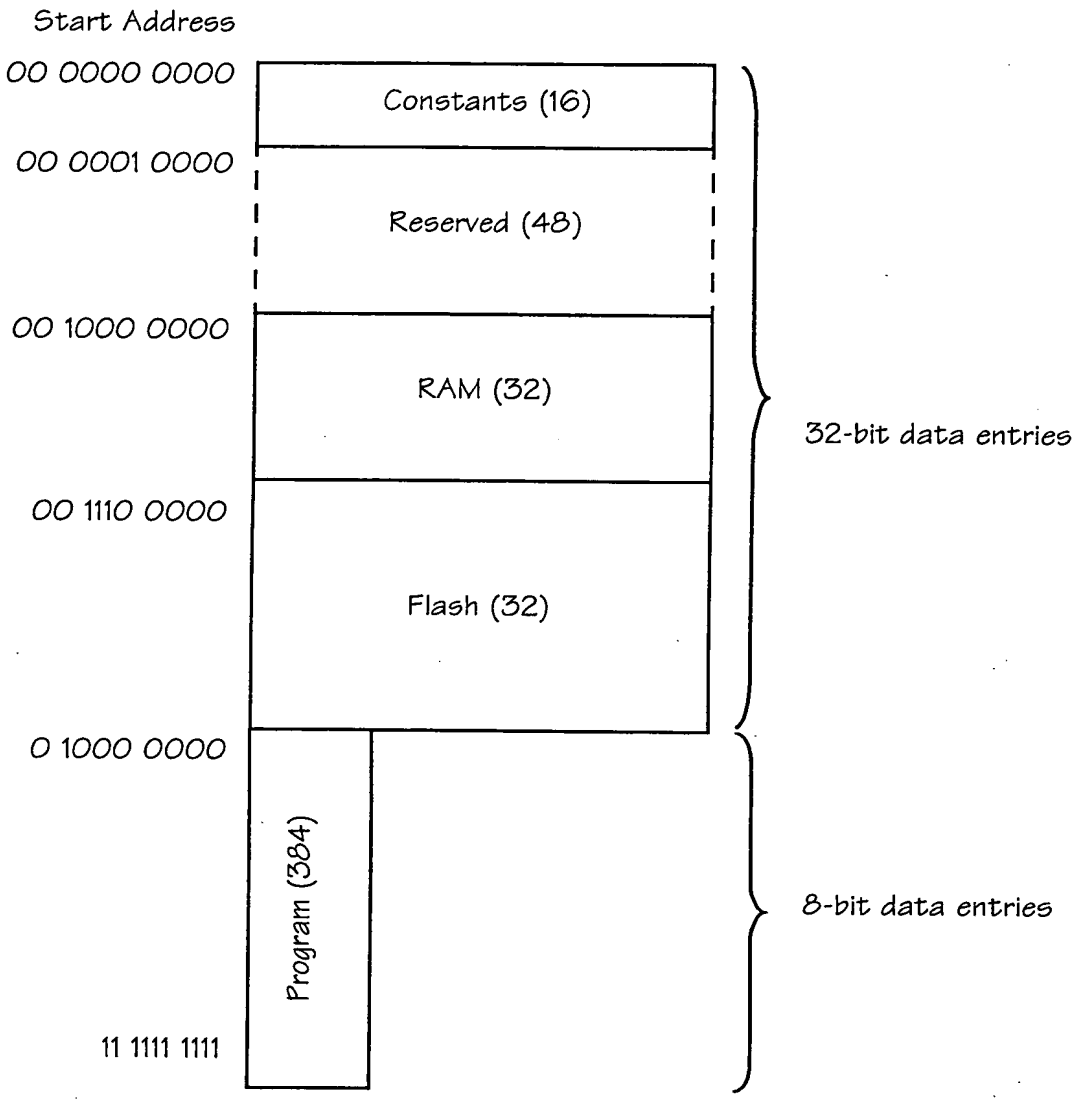

Start Address
00 0000 0000
Constants (16)
00 0001 0000
Reserved (48)
00 1000 0000
RAM (32)
32-bit data entries
00 1110 0000
Flash (32)
0 1000 0000
Program (384)
8-bit data entries
11 1111 1111

FIG. 182

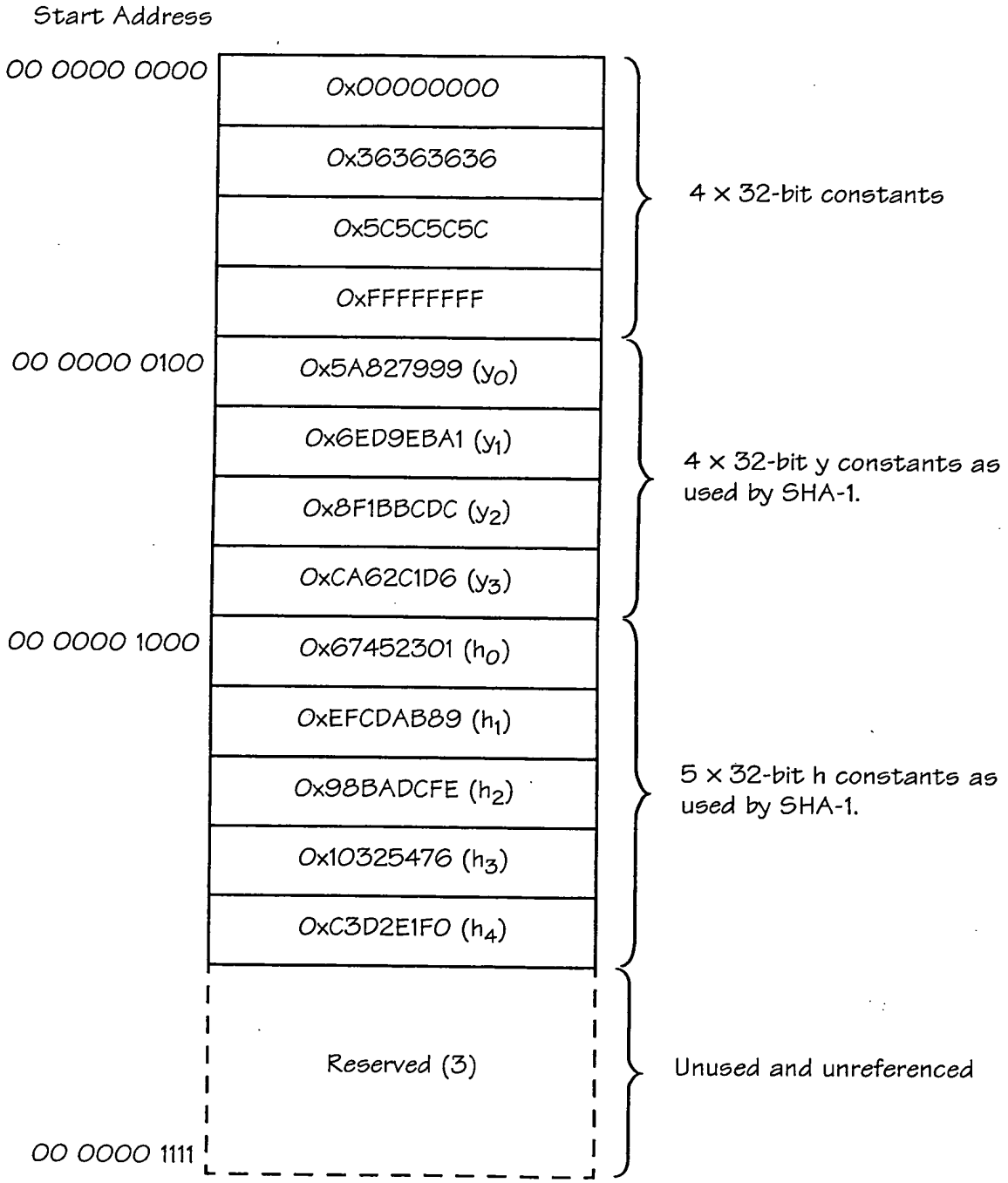

Start Address
00 0000 0000
0x00000000
0x36363636
0x5C5C5C5C
0xFFFFFFFF
4 × 32-bit constants
00 0000 0100
0x5A827999 (y0)
0x6ED9EBA1 (y1)
0x8F1BBCDC (y2)
0xCA62C1D6 (y3)
4 × 32-bit y constants as used by SHA-1.
00 0000 1000
0x67452301 (h0)
0xEFCDAB89 (h1)
0x98BADCFE (h2)
0x10325476 (h3)
0xC3D2E1F0 (h4)
5 × 32-bit h constants as used by SHA-1.
Reserved (3)
Unused and unreferenced
00 0000 1111

FIG. 183

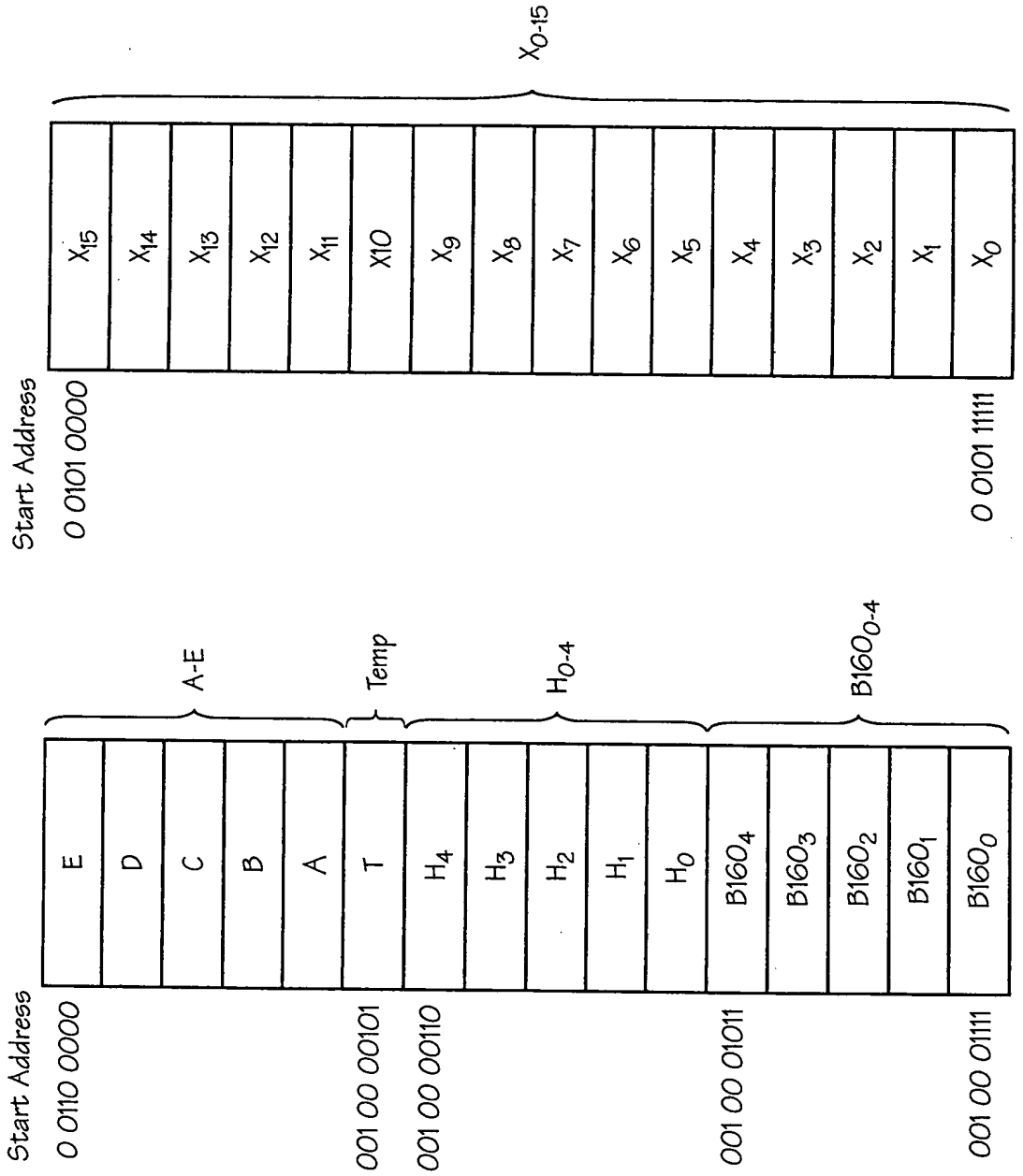

Start Address
0 0110 0000
E
D
C
B
A
A-E
001 00 00101
T
Temp
001 00 00110
H4
H3
H2
H1
H0
H0-4
001 00 01011
B160 4
B160 3
B160 2
B160 1
B160 0
B160 0-4
001 00 01111
Start Address
0 0101 0000
X15
X14
X13
X12
X11
X10
X9
X8
X7
X6
X5
X4
X3
X2
X1
X0
X0-15
0 0101 11111

FIG. 184

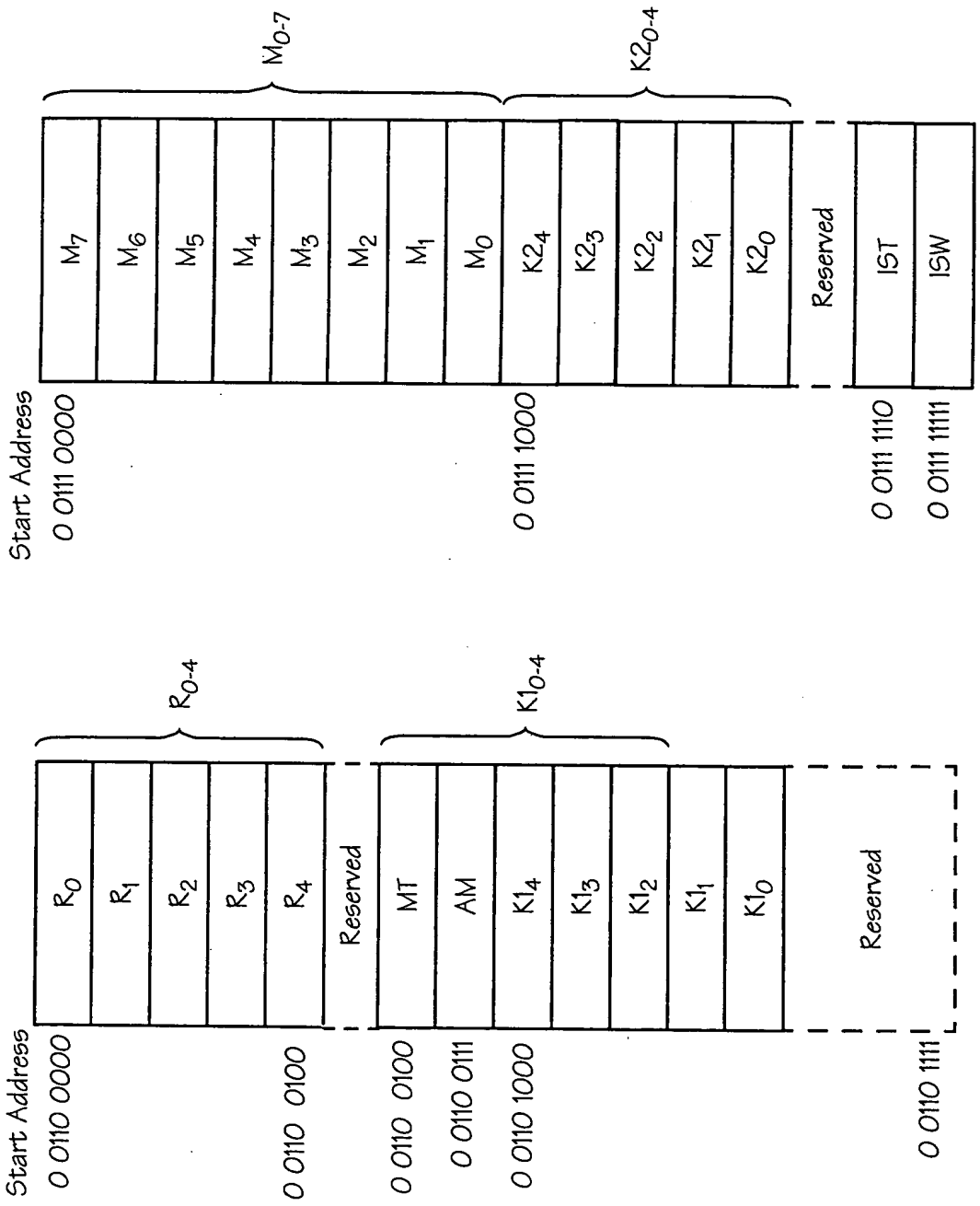

Start Address
0 0110 0000
0 0110 0100
0 0110 0111
0 0110 1000
0 0110 1111
R0-4
R0
R1
R2
R3
R4
Reserved
K1 0-4
MT
AM
K14
K13
K12
K11
K10
Reserved
M0-7
M7
M6
M5
M4
M3
M2
M1
M0
K2 0-4
K24
K23
K22
K21
K20
Reserved
IST
ISW
0 0111 0000
0 0111 1000
0 0111 1110
0 0111 1111

FIG. 185

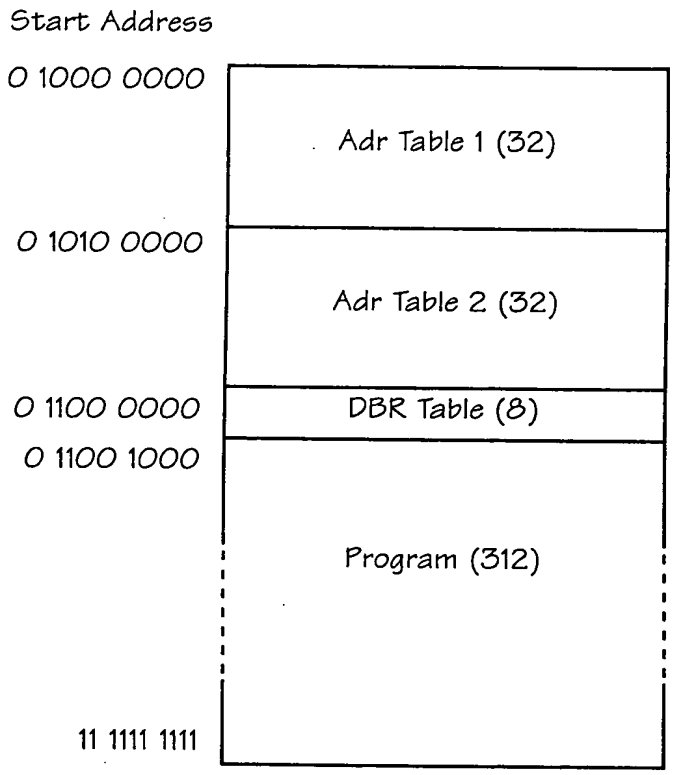

Start Address
0 1000 0000
Adr Table 1 (32)
0 1010 0000
Adr Table 2 (32)
0 1100 0000
DBR Table (8)
0 1100 1000
Program (312)
11 1111 1111

FIG. 186

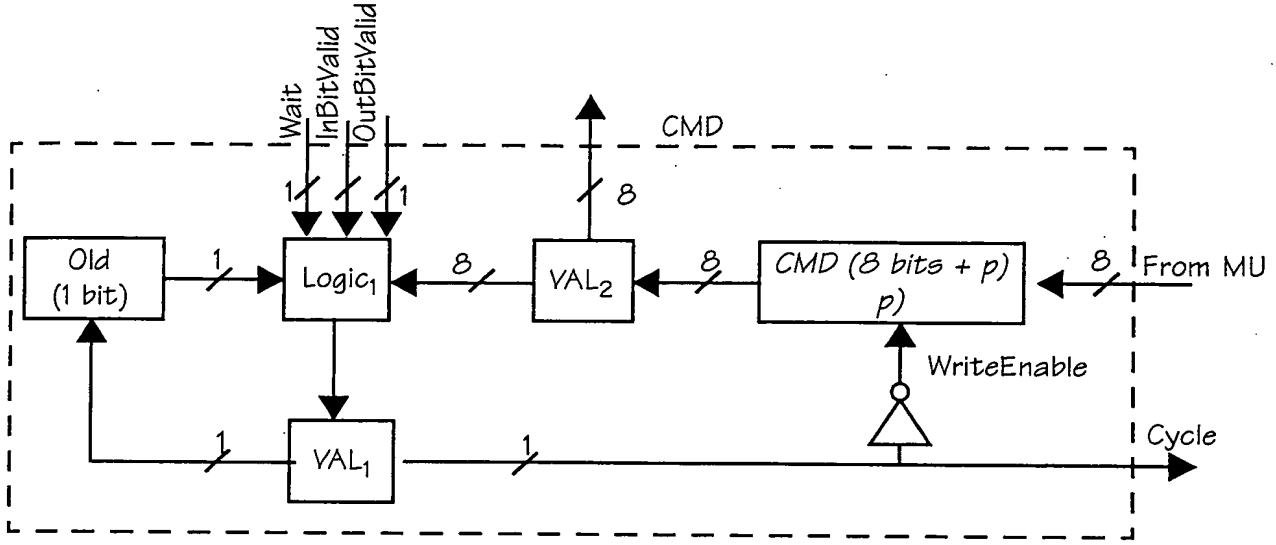


FIG. 187

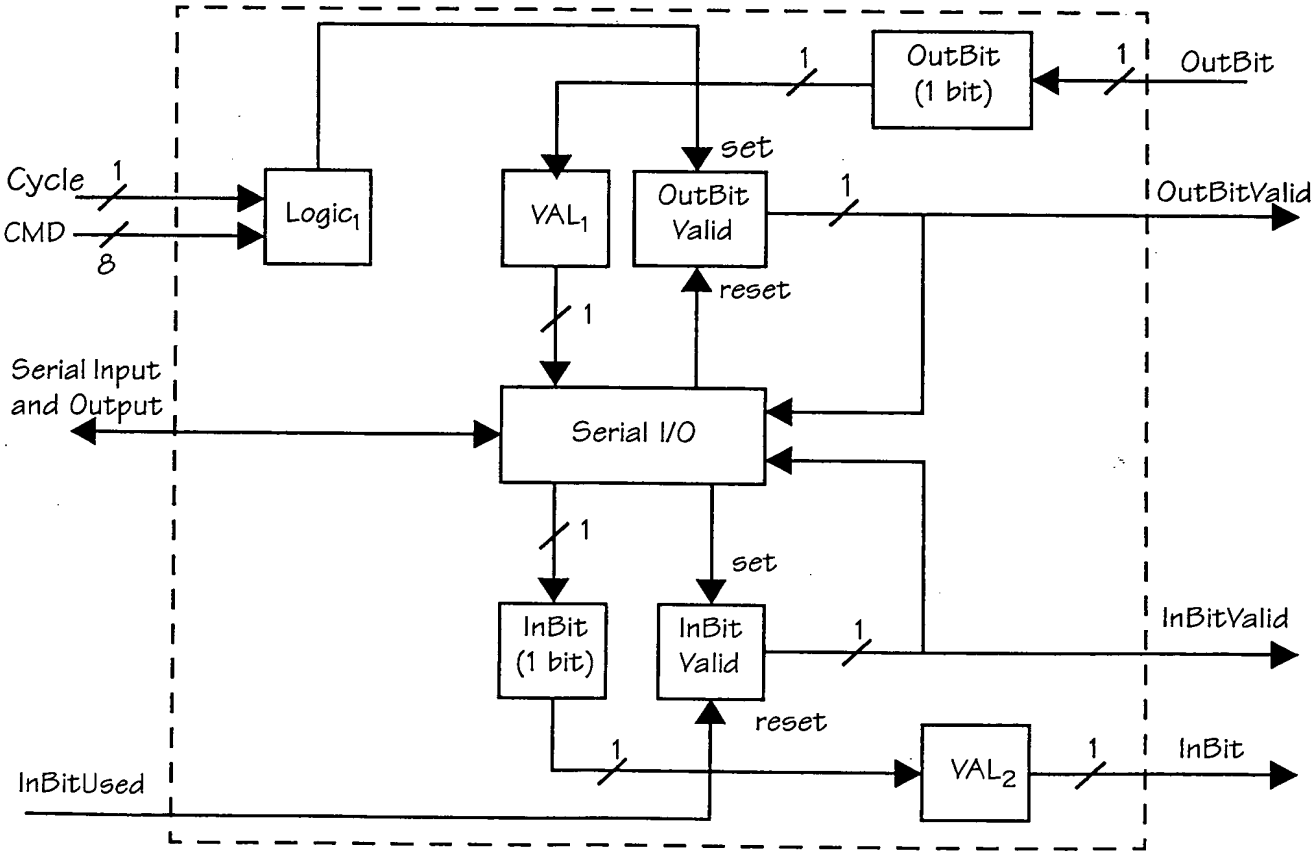


FIG. 188

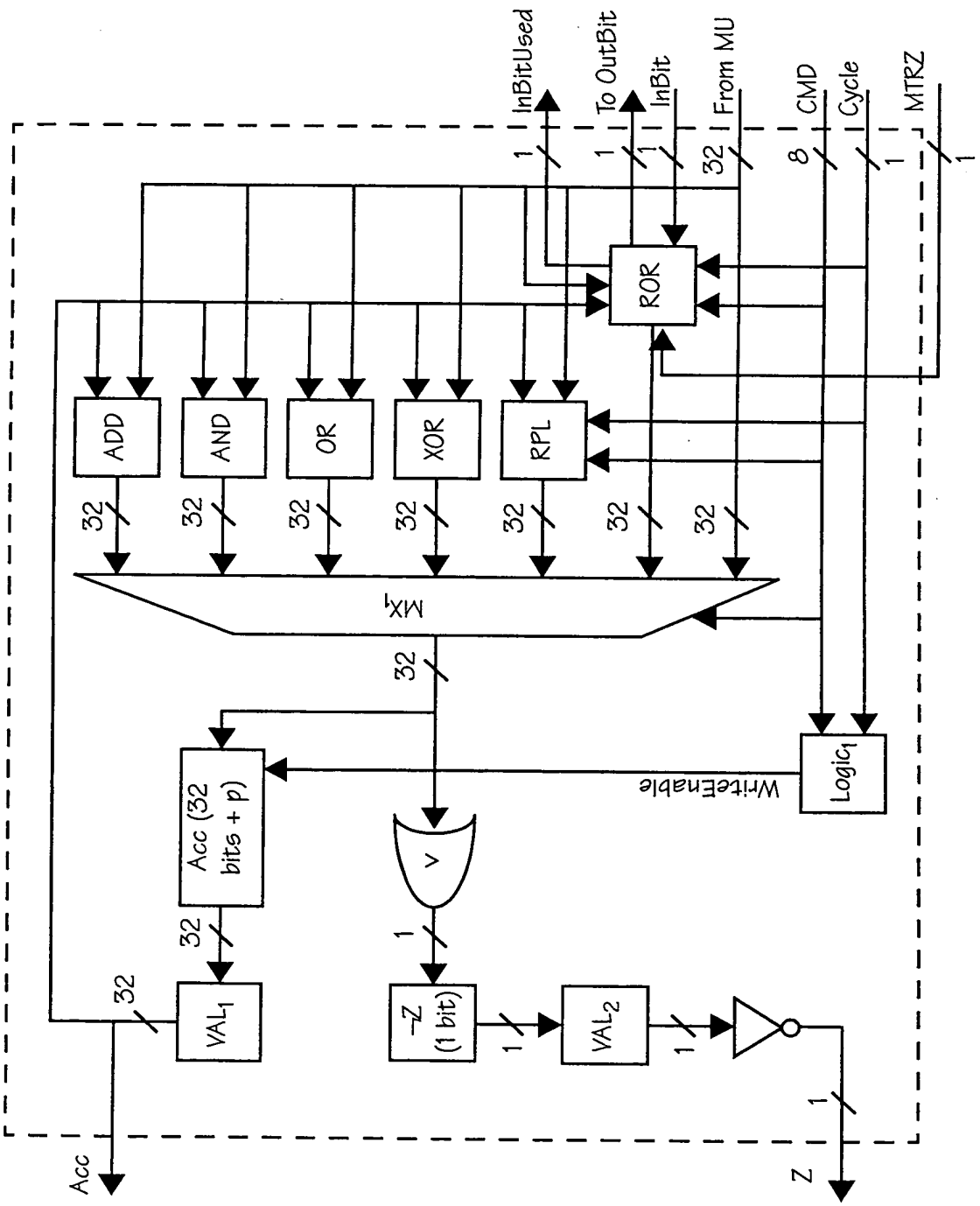

InBitUsed
To OutBit
InBit
From MU
CMD
Cycle
MTRZ
ROR
ADD
AND
OR
XOR
RPL
MX1
Acc (32 bits + p)
V
Logic1
WriteEnable
VAL1
-Z (1 bit)
VAL2
Acc
Z

FIG. 189

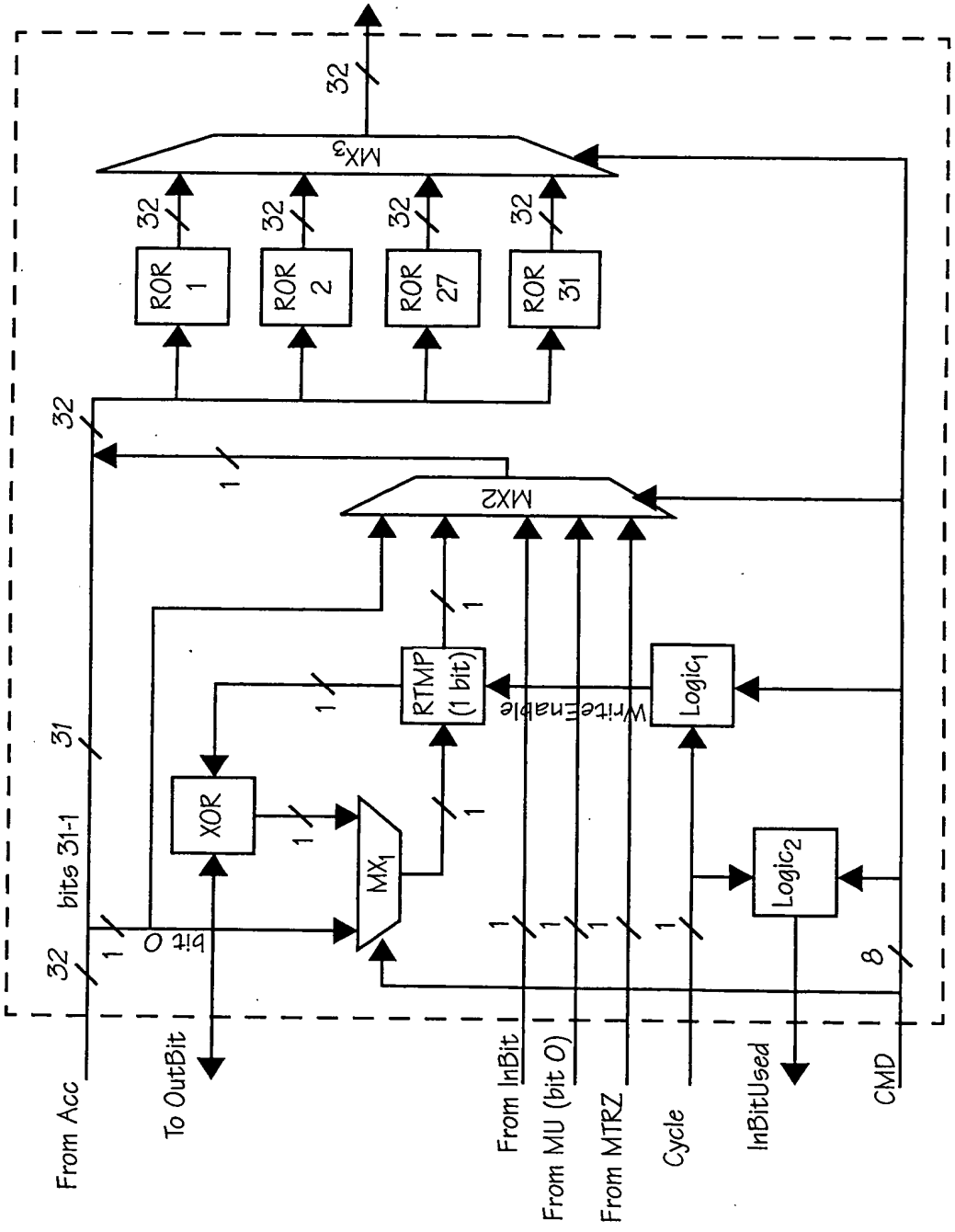

From Acc
To OutBit
From InBit
From MU (bit 0)
From MTRZ
Cycle
InBitUsed
CMD
32
bits 31-1
31
bit 0
1
XOR
MX1
RTMP (1 bit)
WriteEnable
Logic1
Logic2
MX2
MX3
ROR 1
ROR 2
ROR 27
ROR 31
8

FIG. 190

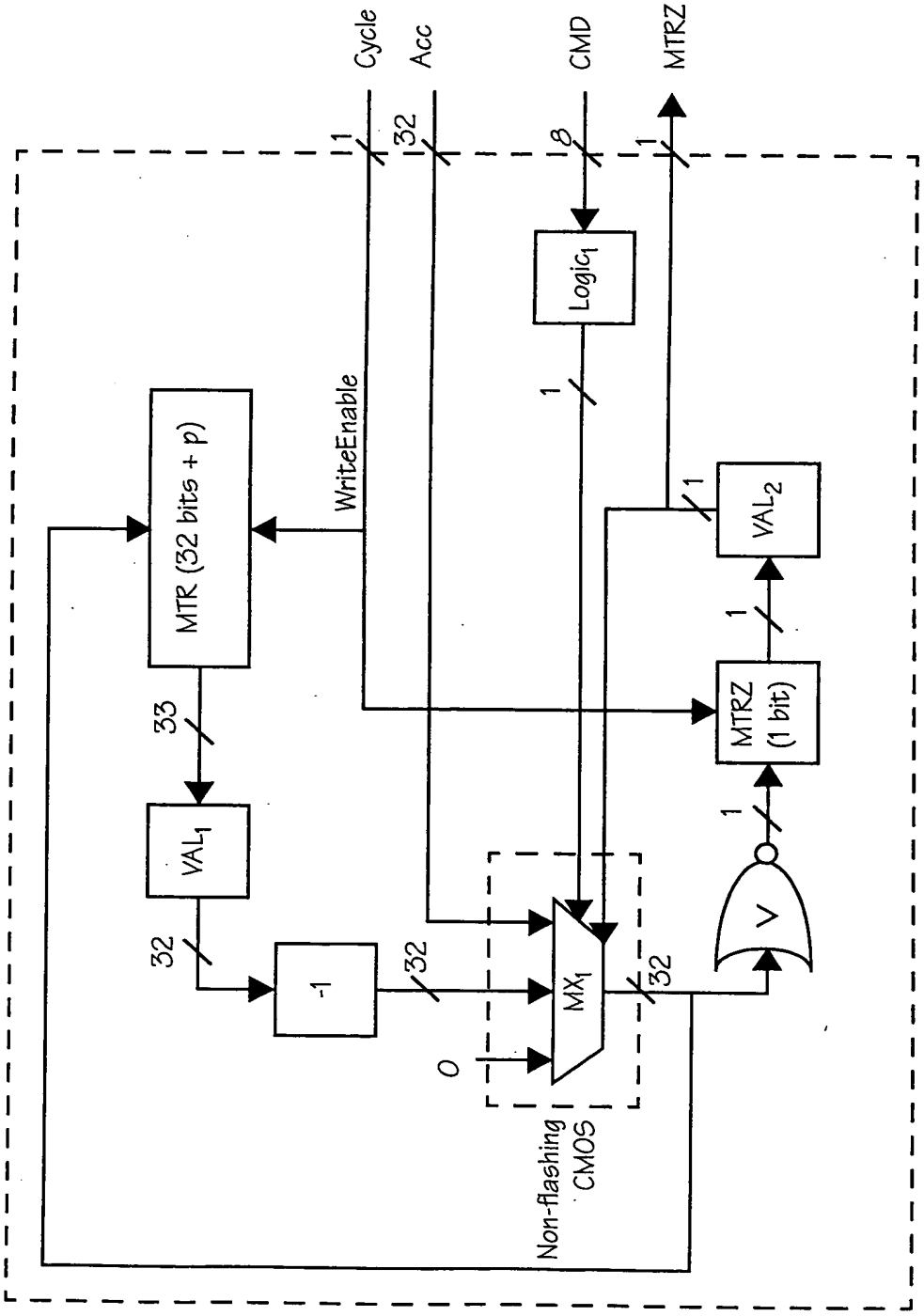

Cycle
Acc
CMD
MTRZ
1
32
8
1
Logic1
1
WriteEnable
MTR (32 bits + p)
33
VAL1
32
-1
32
MX1
32
0
Non-flashing CMOS
1
VAL2
1
MTRZ (1 bit)
1
V

FIG. 191

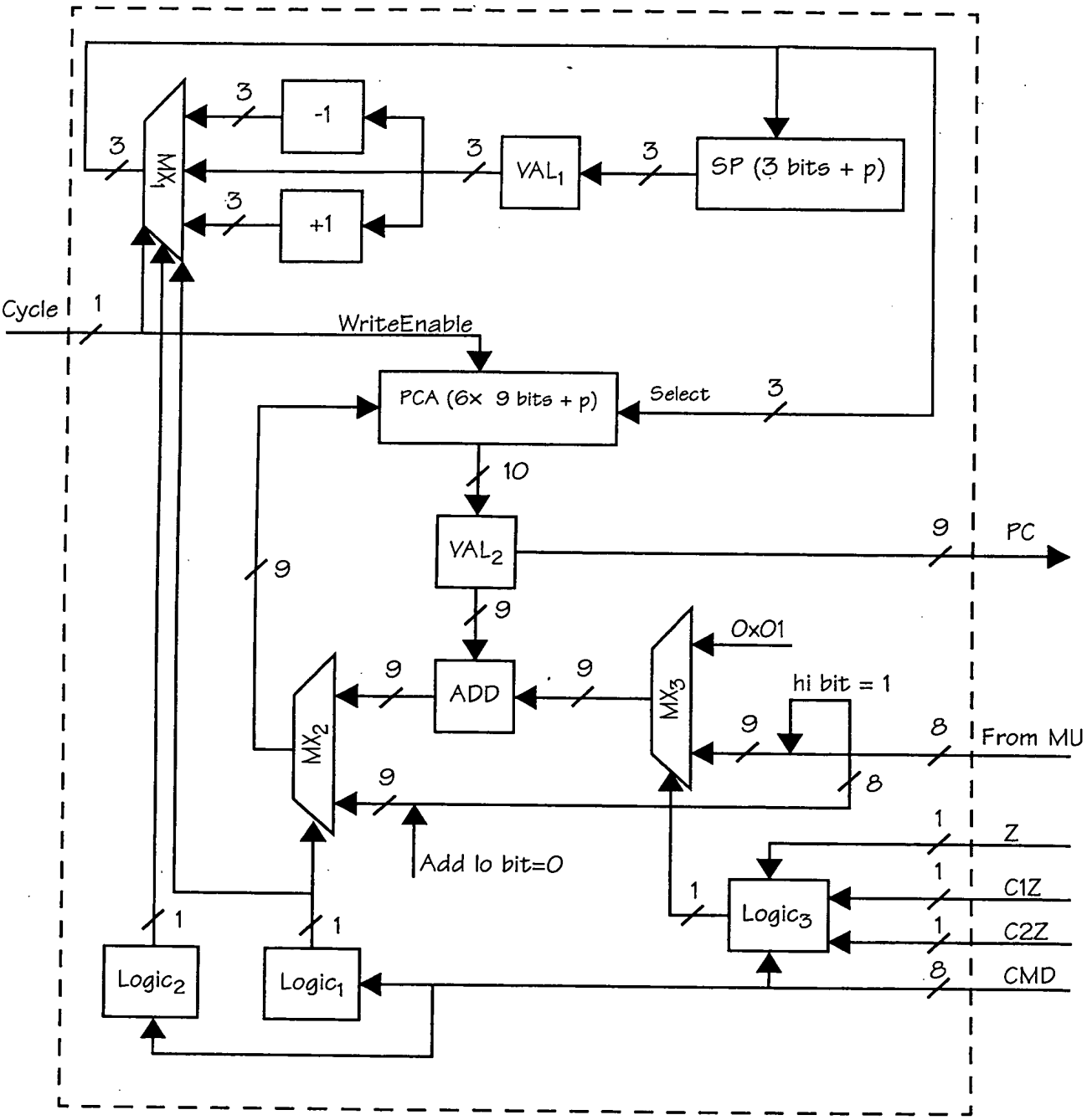

MX1
-1
+1
VAL1
SP (3 bits + p)
Cycle
WriteEnable
PCA (6x 9 bits + p)
Select
VAL2
PC
ADD
MX2
MX3
0x01
hi bit = 1
From MU
Add lo bit=0
Logic3
Logic2
Logic1
Z
C1Z
C2Z
CMD

FIG. 192

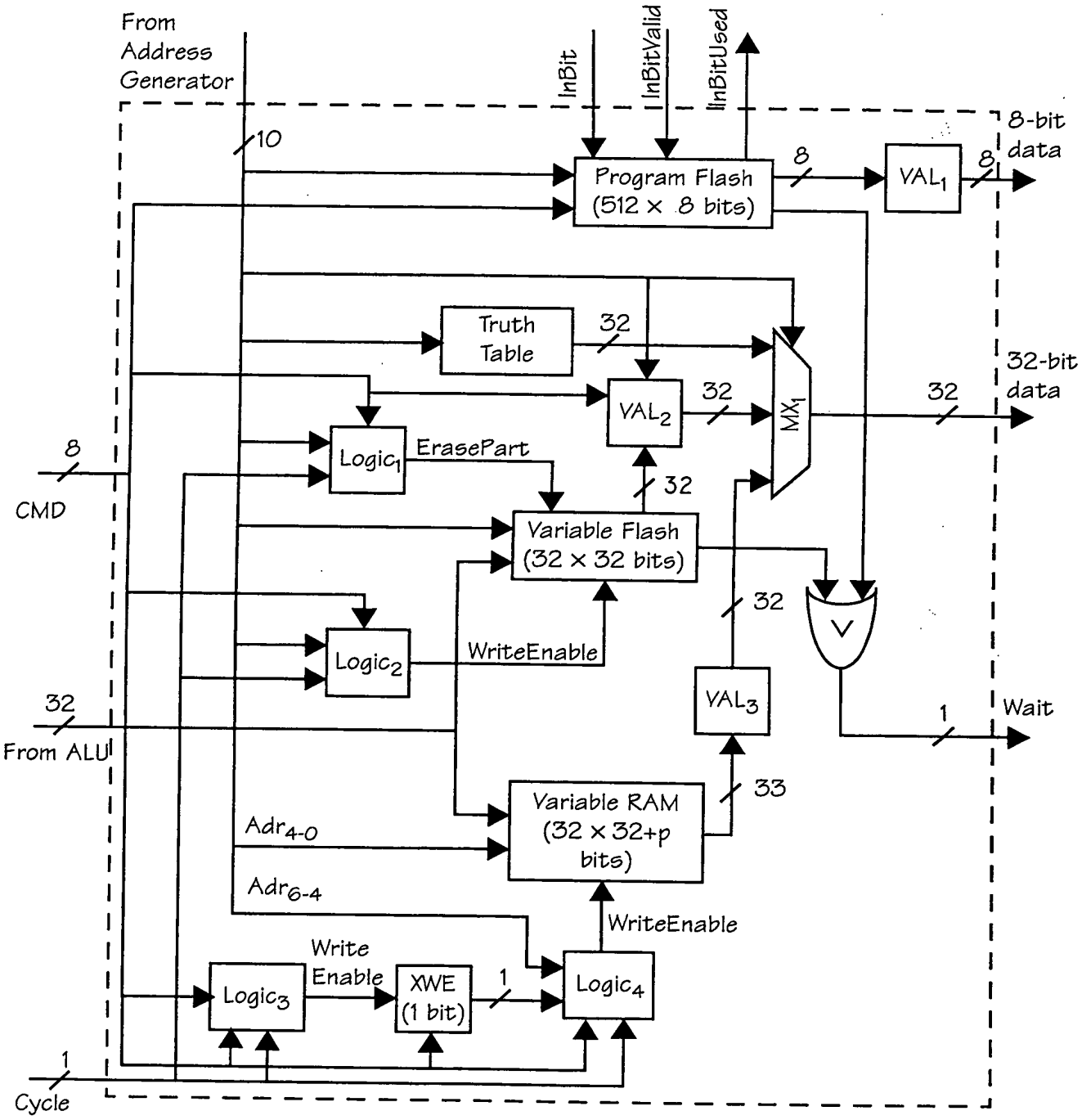

From Address Generator
InBit
InBitValid
InBitUsed
10
8-bit data
Program Flash (512 x 8 bits)
8
VAL1
Truth Table
32
32-bit data
VAL2
MX1
Logic1
ErasePart
CMD
Variable Flash (32 x 32 bits)
Logic2
WriteEnable
V
VAL3
From ALU
Wait
Variable RAM (32 x 32+p bits)
33
Adr4-0
Adr6-4
WriteEnable
Write Enable
Logic3
XWE (1 bit)
Logic4
Cycle

FIG. 193

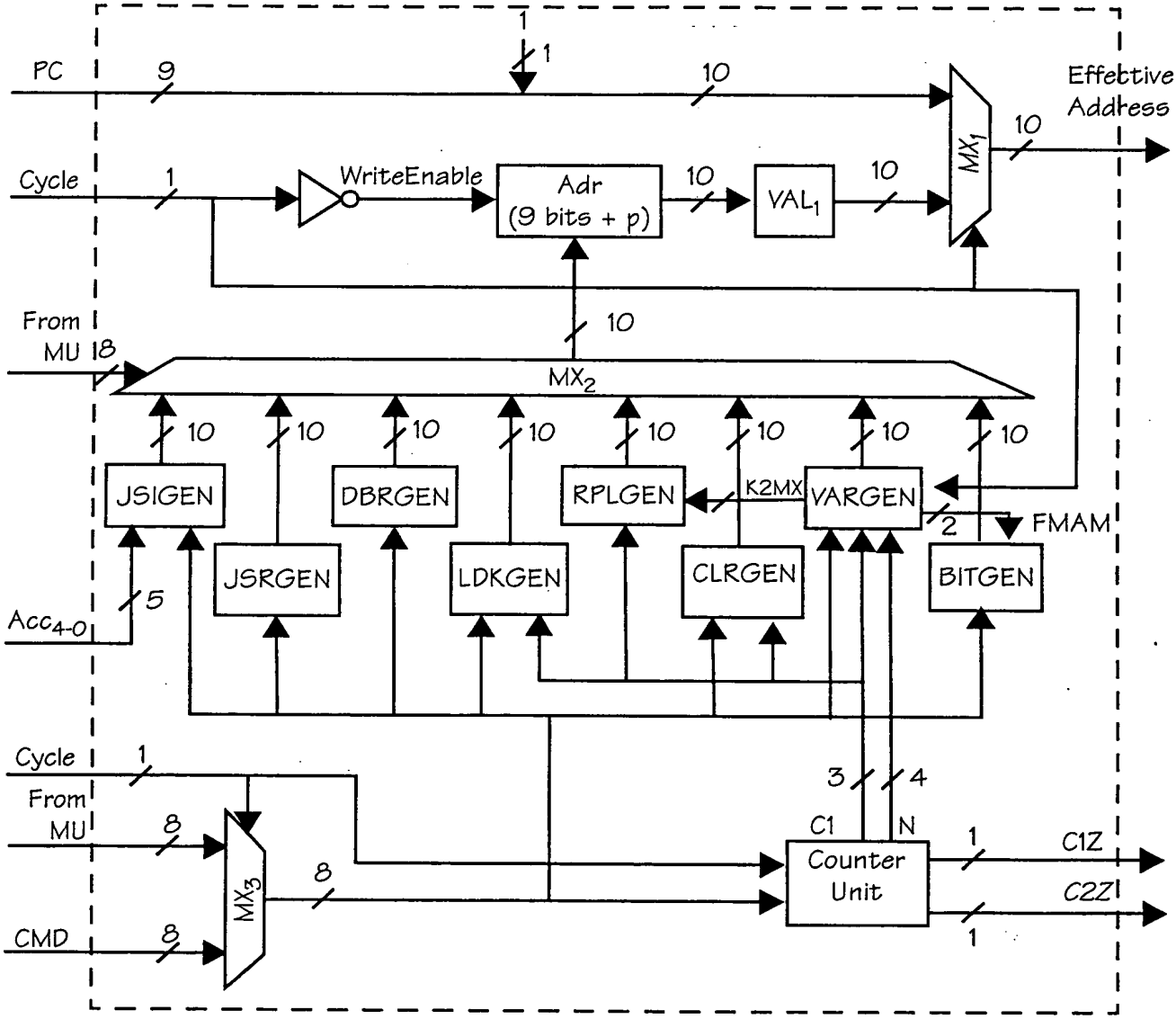

PC
9
1
1
10
Effective Address
10
MX1
Cycle
1
WriteEnable
Adr (9 bits + p)
10
VAL1
10
From MU
18
10
MX2
10
10
10
10
10
10
10
10
JSIGEN
DBRGEN
RPLGEN
K2MX
VARGEN
2
FMAM
JSRGEN
LDKGEN
CLRGEN
BITGEN
Acc4-0
5
Cycle
1
From MU
8
CMD
8
MX3
8
3
4
C1
N
Counter Unit
1
C1Z
C2Z
1

FIG. 194

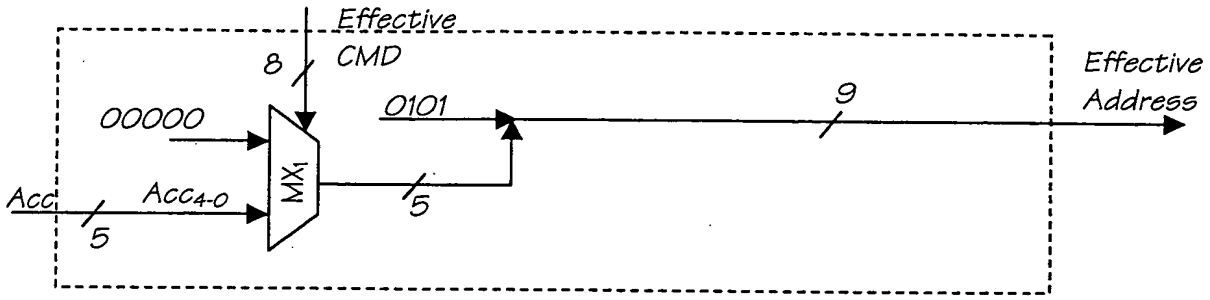


FIG. 195

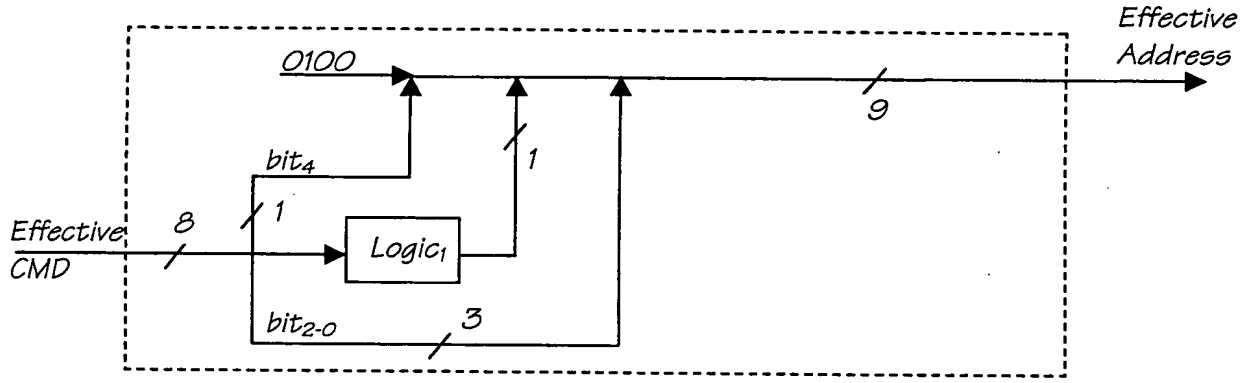


FIG. 196

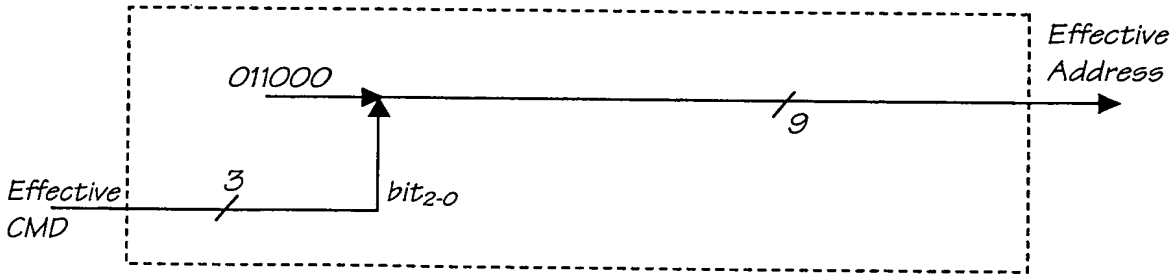


FIG. 197

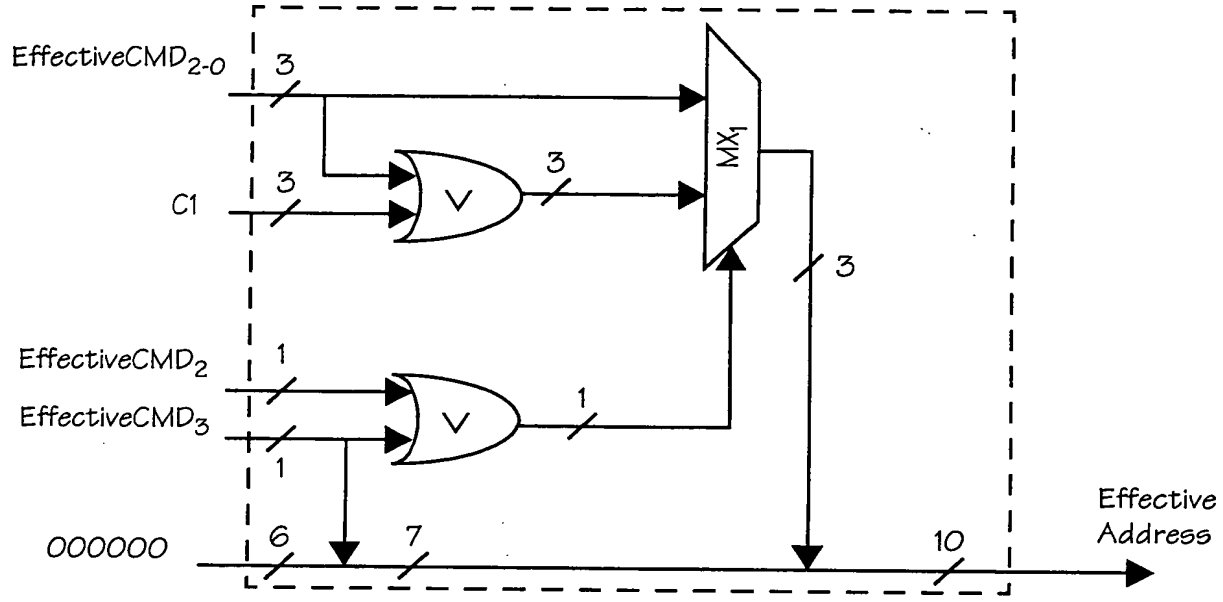


FIG. 198

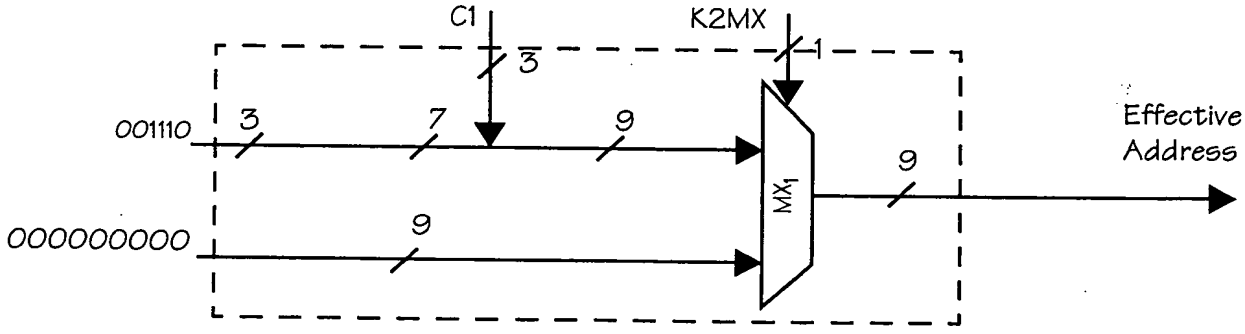


FIG. 199

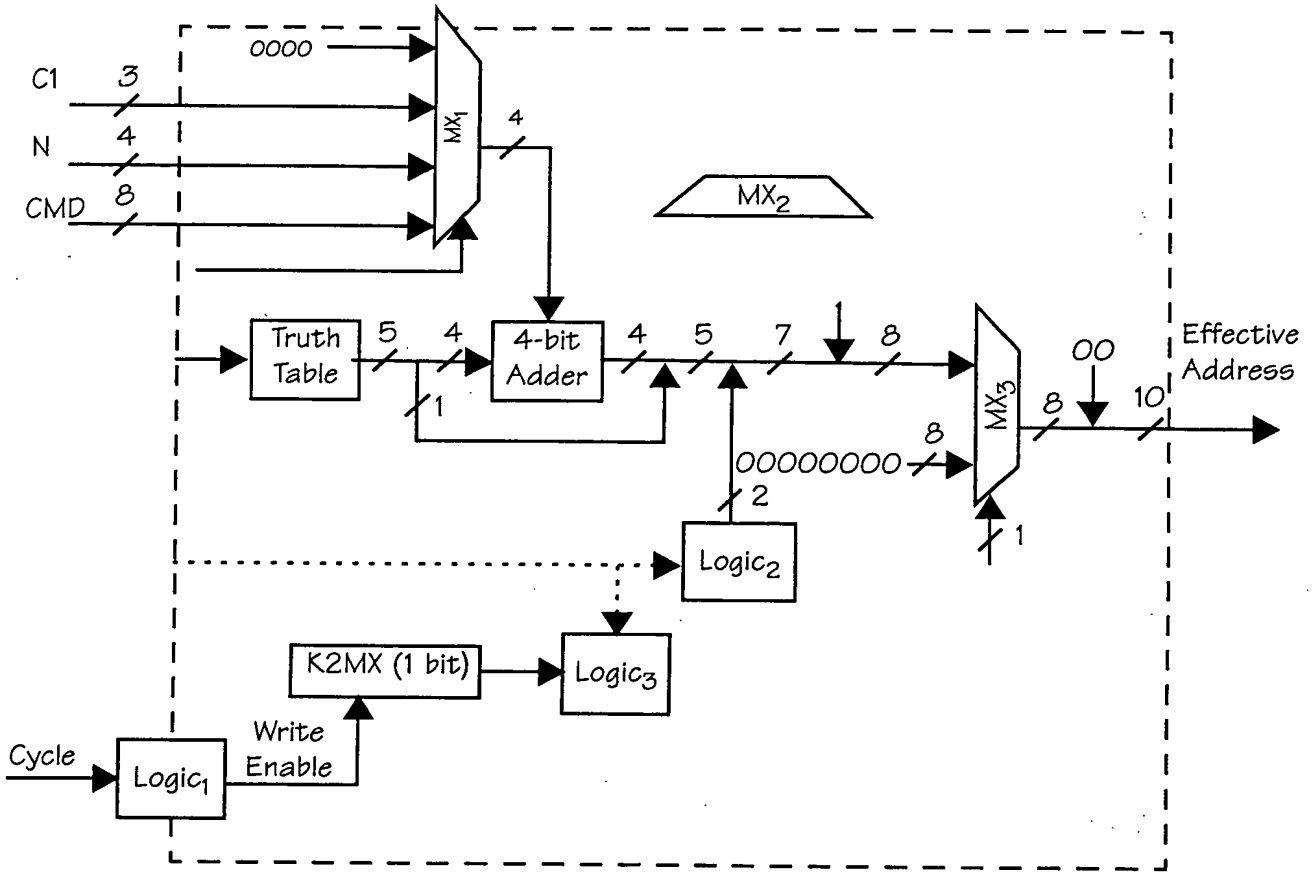


FIG. 200

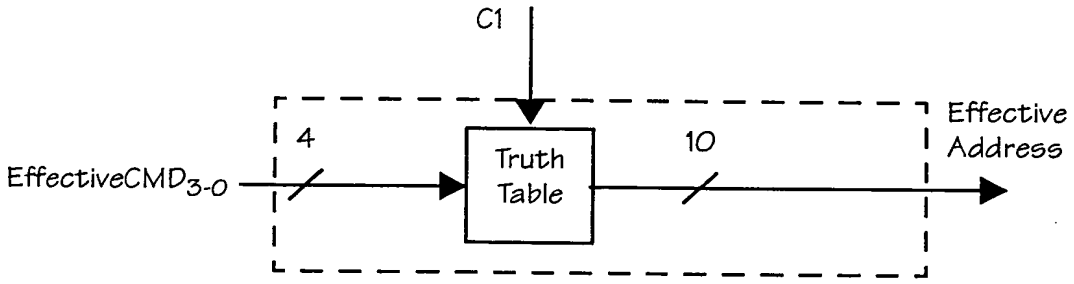


FIG. 201

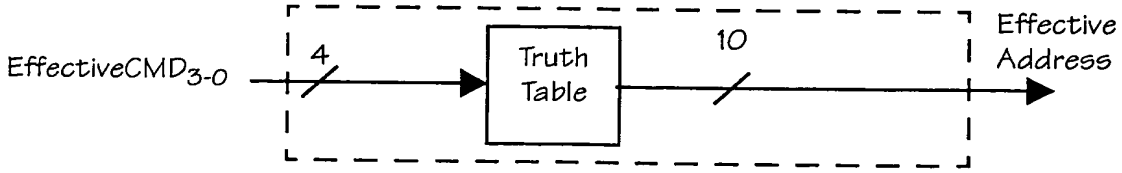


FIG. 202

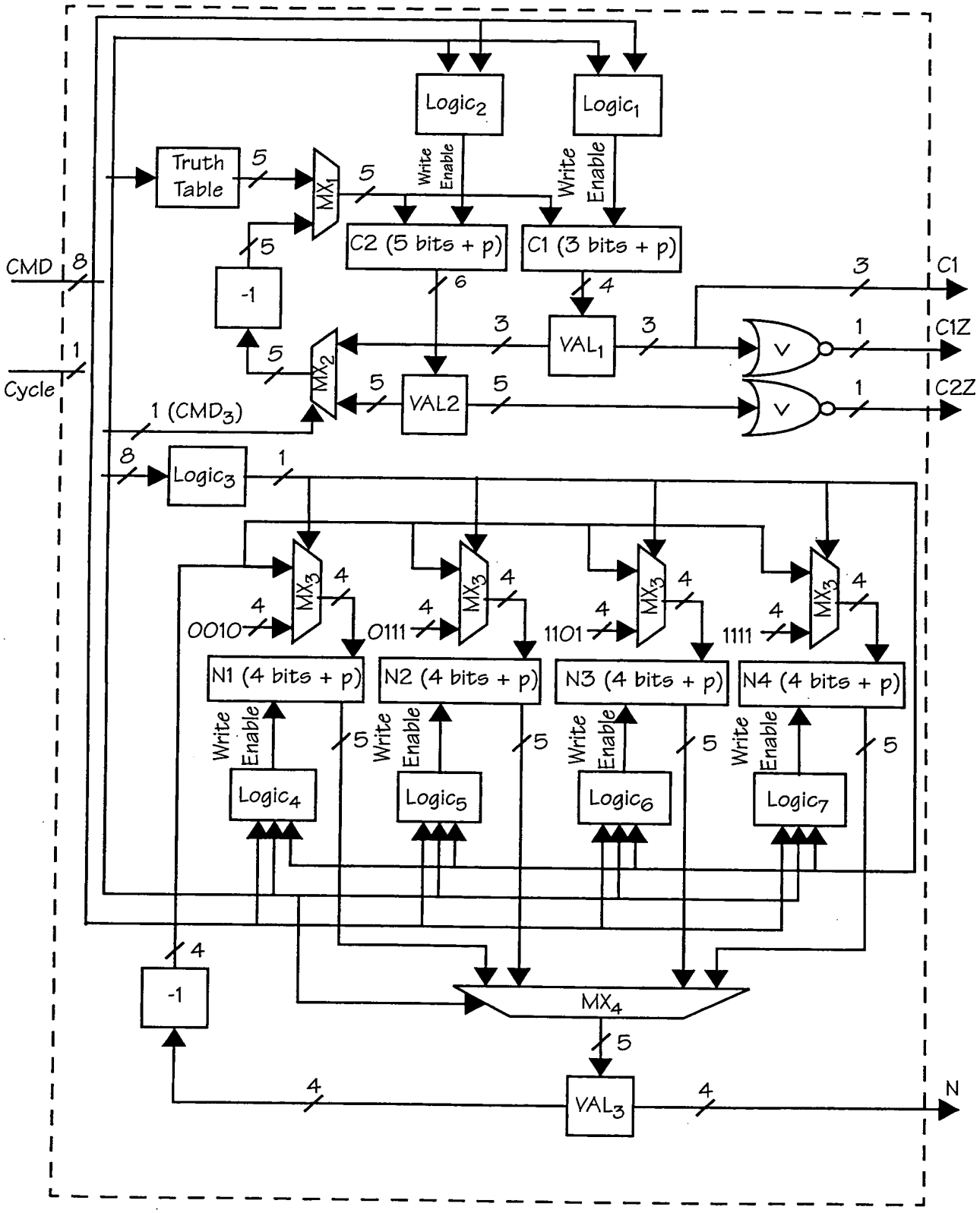

Logic2
Logic1
Truth Table
5
MX1
5
Write Enable
Write Enable
C2 (5 bits + p)
C1 (3 bits + p)
CMD
8
5
-1
6
4
3
C1
3
VAL1
3
V
1
C1Z
1
Cycle
5
MX2
5
VAL2
5
V
1
C2Z
1 (CMD3)
8
Logic3
1
MX3
4
4
0010
0111
1101
1111
N1 (4 bits + p)
N2 (4 bits + p)
N3 (4 bits + p)
N4 (4 bits + p)
Write Enable
5
Logic4
Logic5
Logic6
Logic7
4
-1
MX4
5
4
VAL3
4
N

FIG. 203

705

DATA TYPE	BITS
Factory Code	16
Batch Number	32
Serial Number	48
Manufacturing Date	16
Media Length	24
Media Type	8
Preprinted Media Length	16
Cyan Ink Viscosity	8
Magenta Ink Viscosity	8
Yellow Ink Viscosity	8
Cyan Drop Volume	8
Magenta Drop Volume	8
Yellow Drop Volume	8
Cyan Ink Color	24
Magenta Ink Color	24
Yellow Ink Color	24
Remaining-media Length Indicator	16
Authentication Key	128
Copyrightable bit pattern	512
Reserved for Camera Use	88
Total	1024

728

FIG. 204

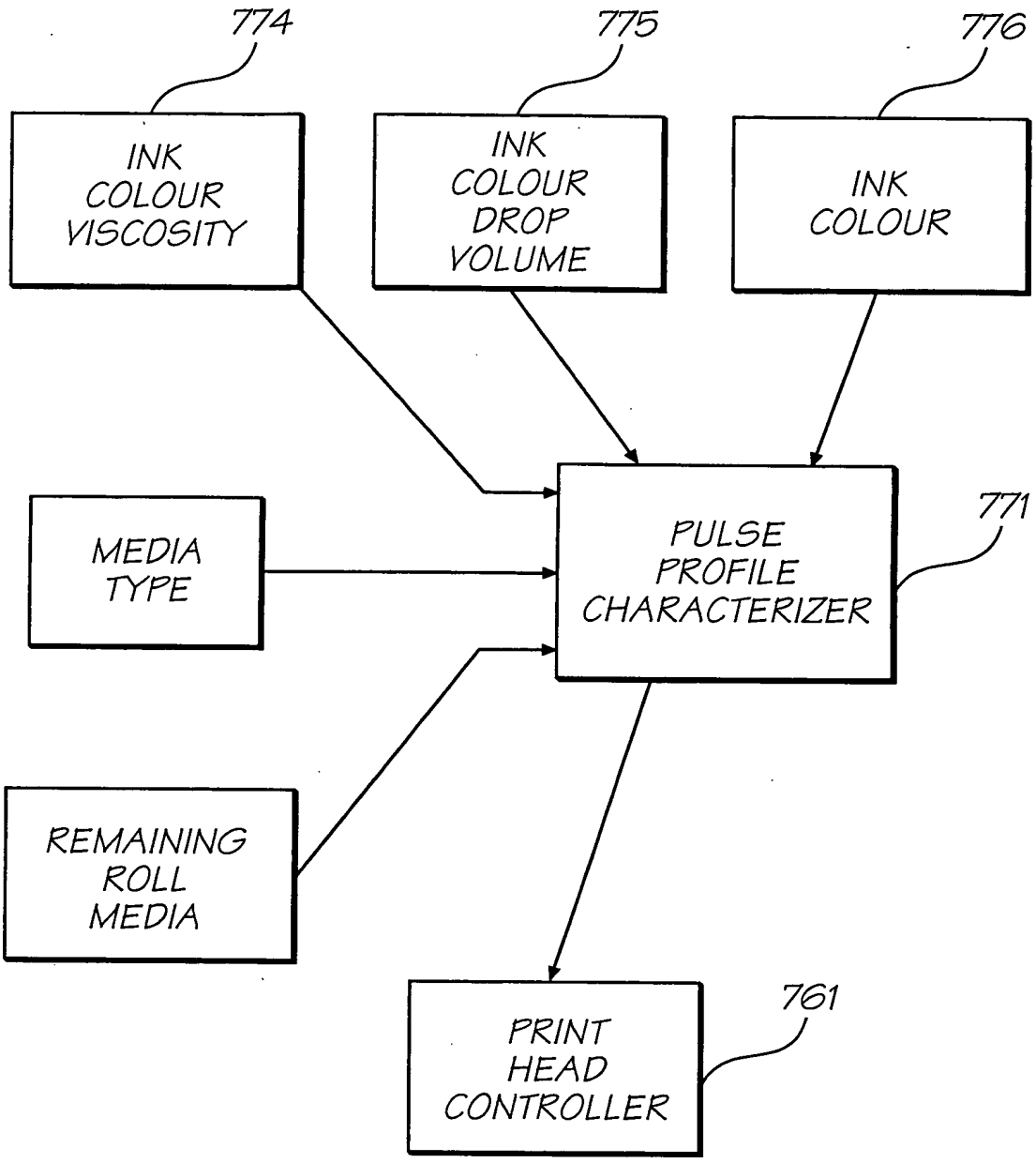

774
775
776
INK COLOUR VISCOSITY
INK COLOUR DROP VOLUME
INK COLOUR
MEDIA TYPE
PULSE PROFILE CHARACTERIZER
771
REMAINING ROLL MEDIA
PRINT HEAD CONTROLLER
761

FIG. 205

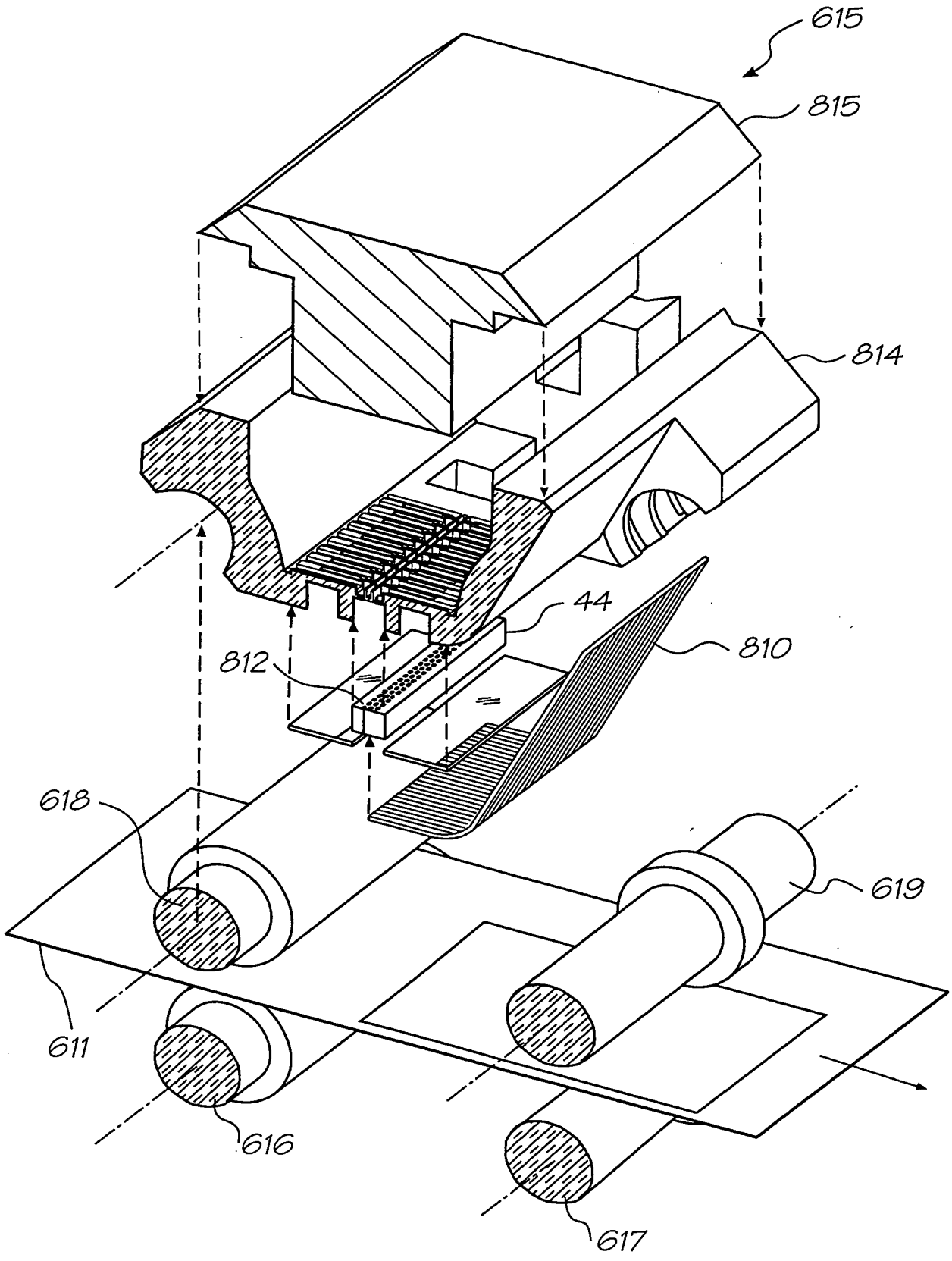


FIG. 206

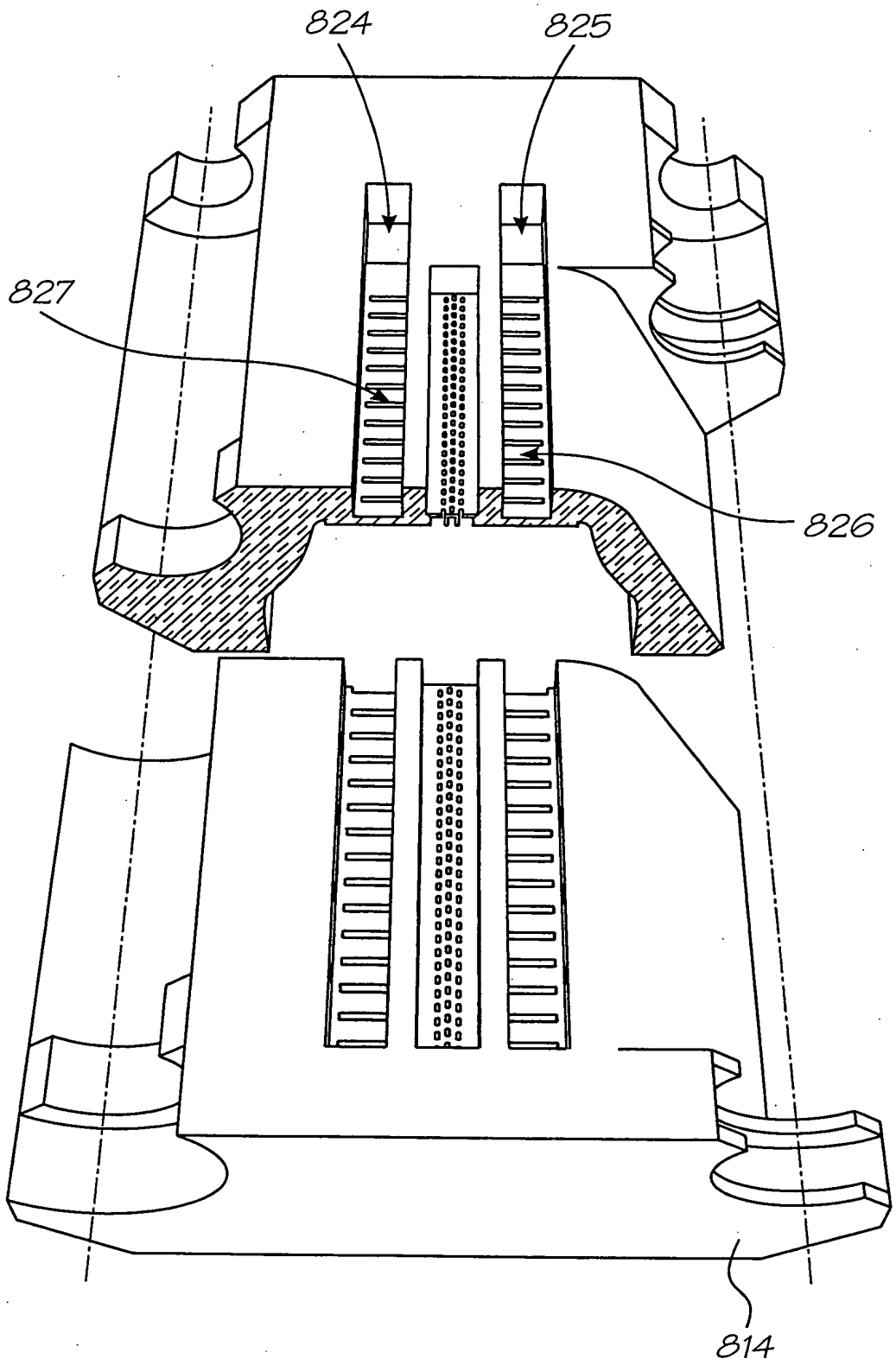


FIG. 207

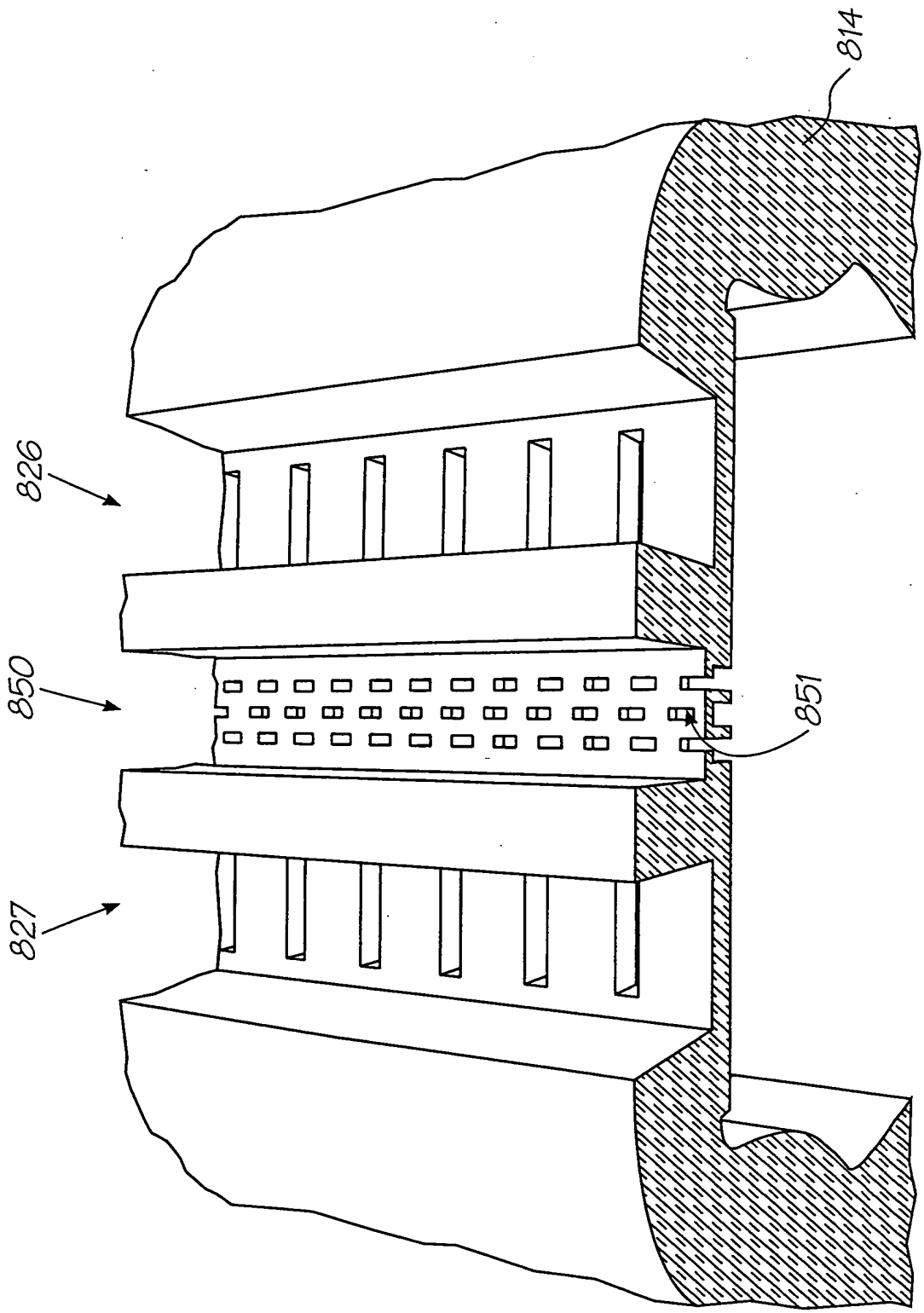


FIG. 208

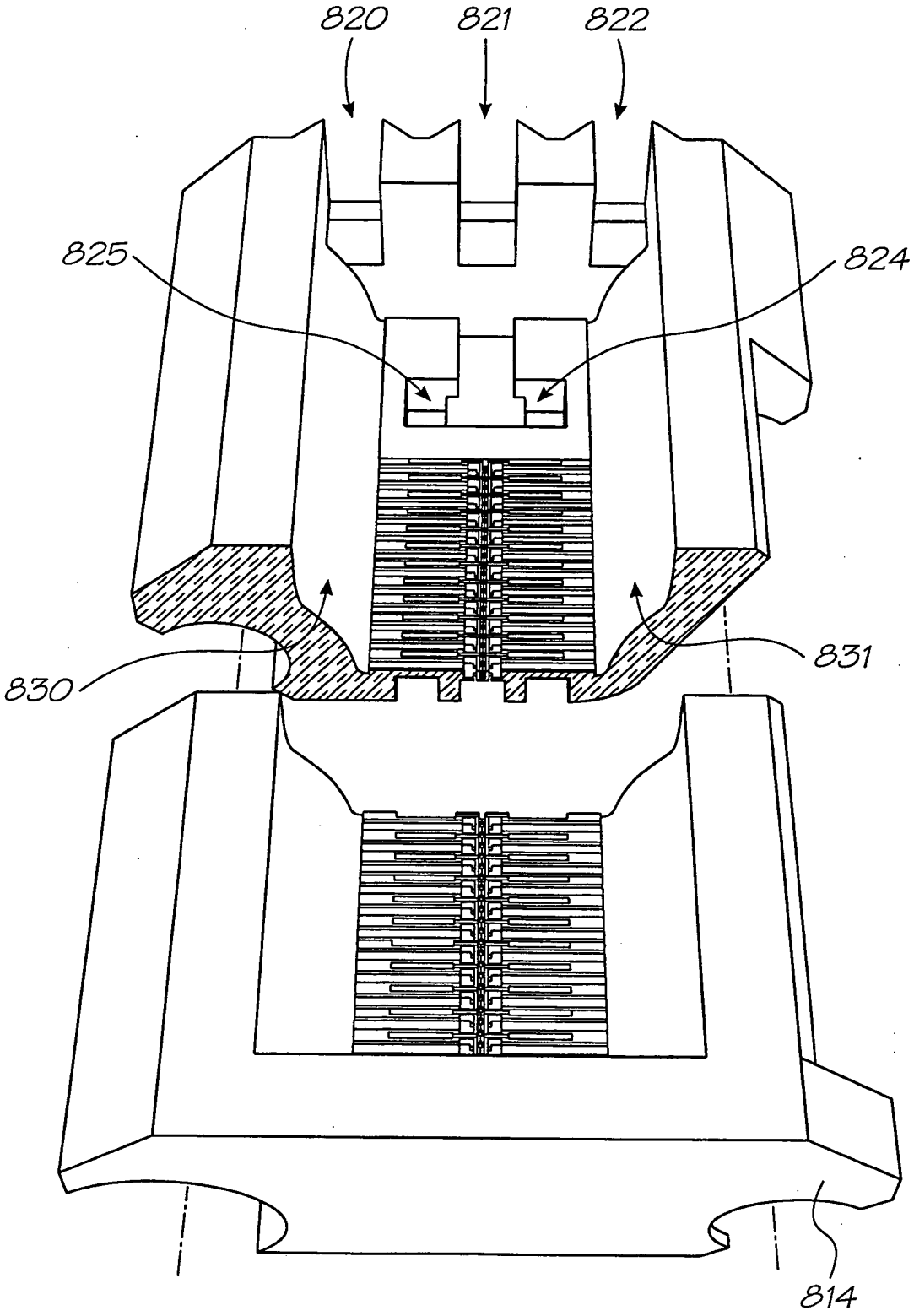


FIG. 209

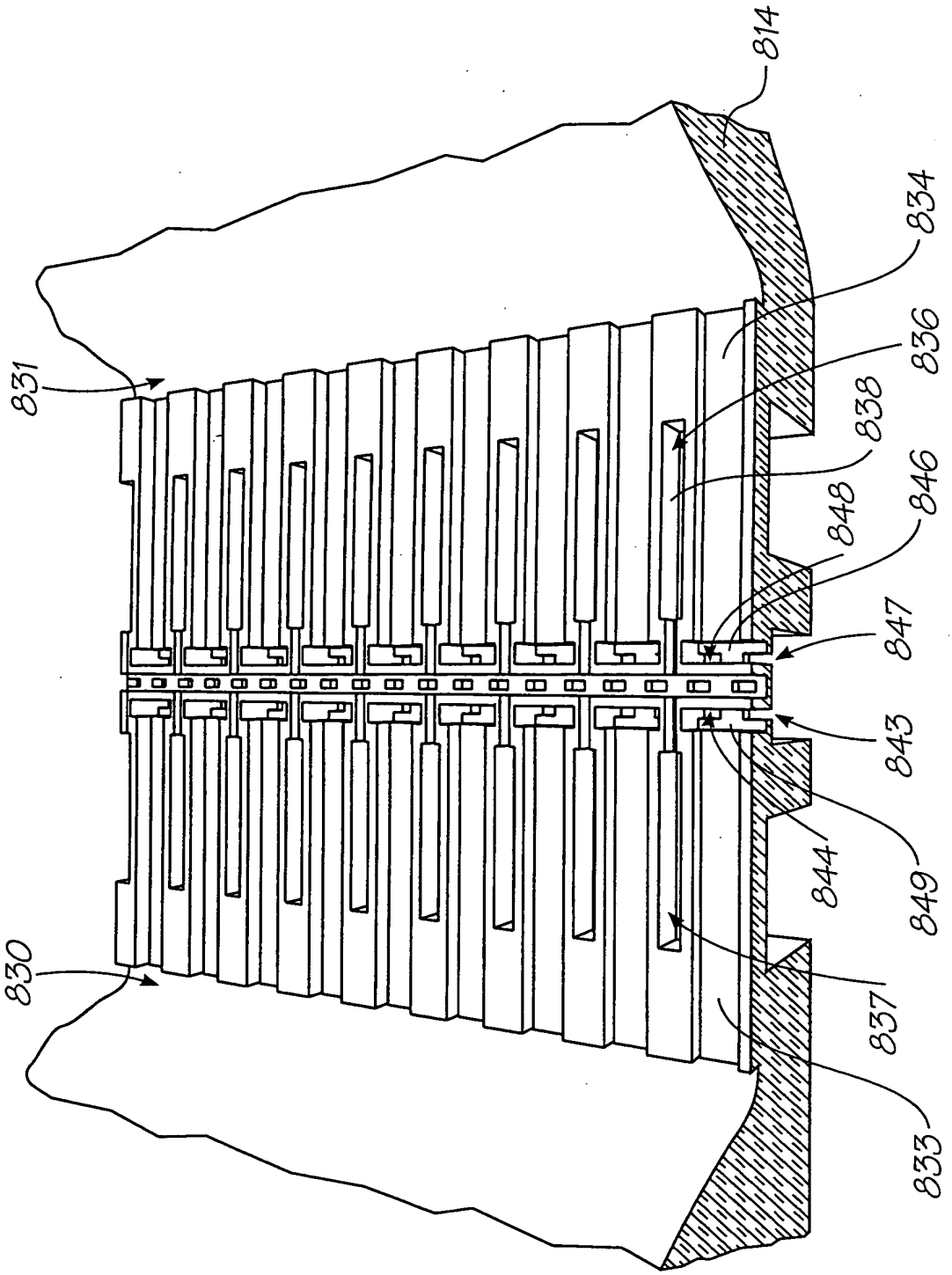


FIG. 210

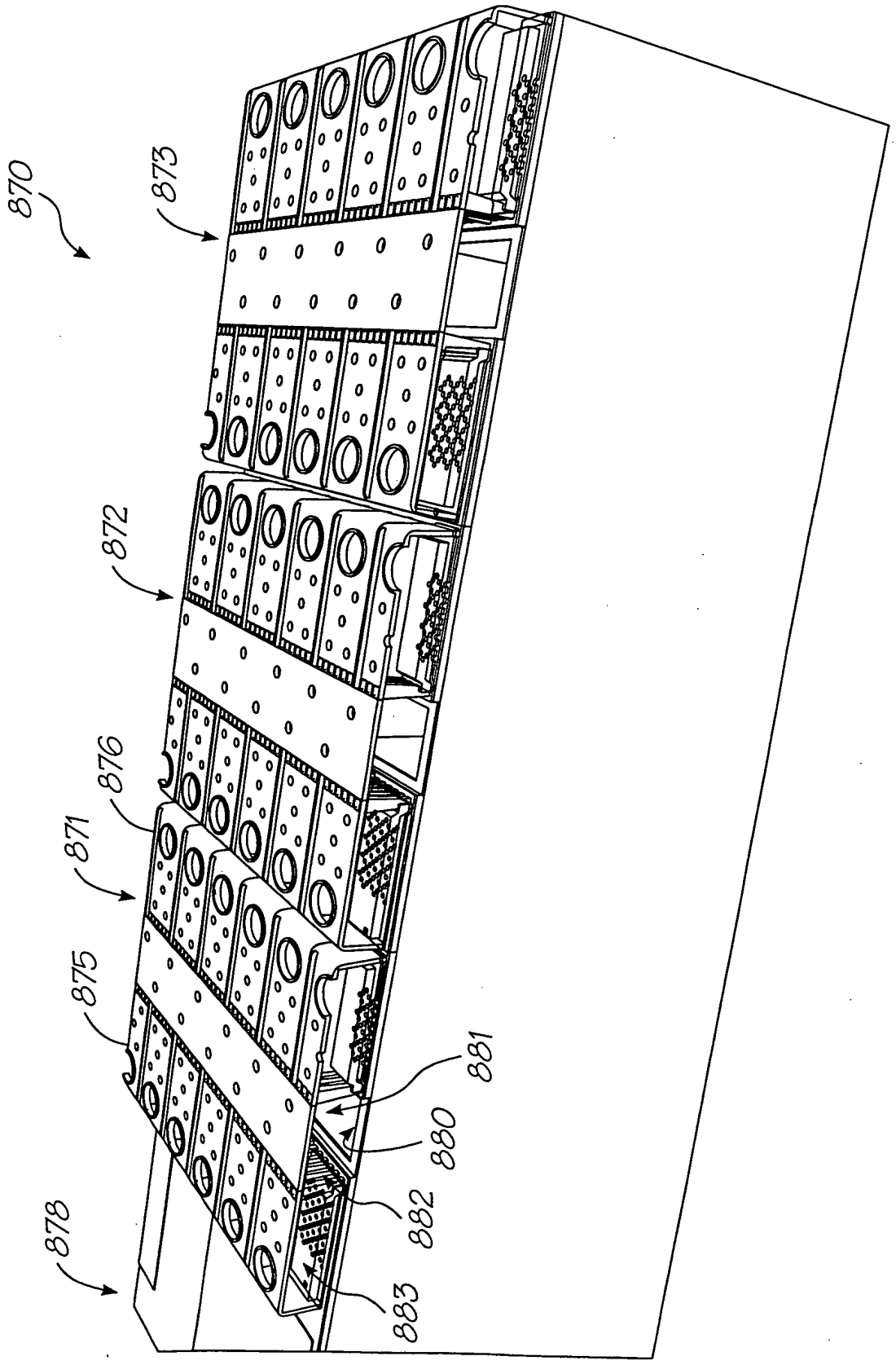


FIG. 211

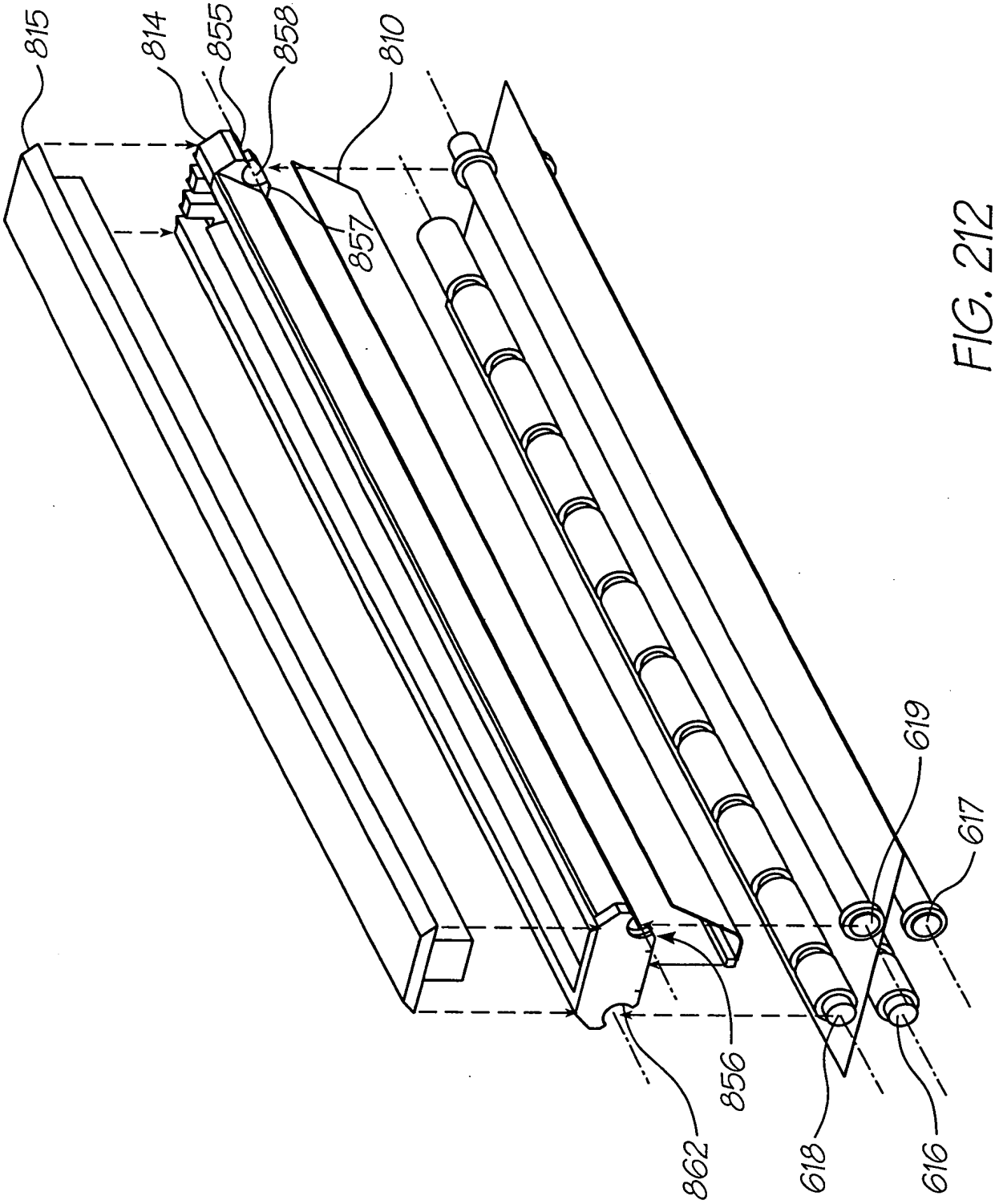


FIG. 212

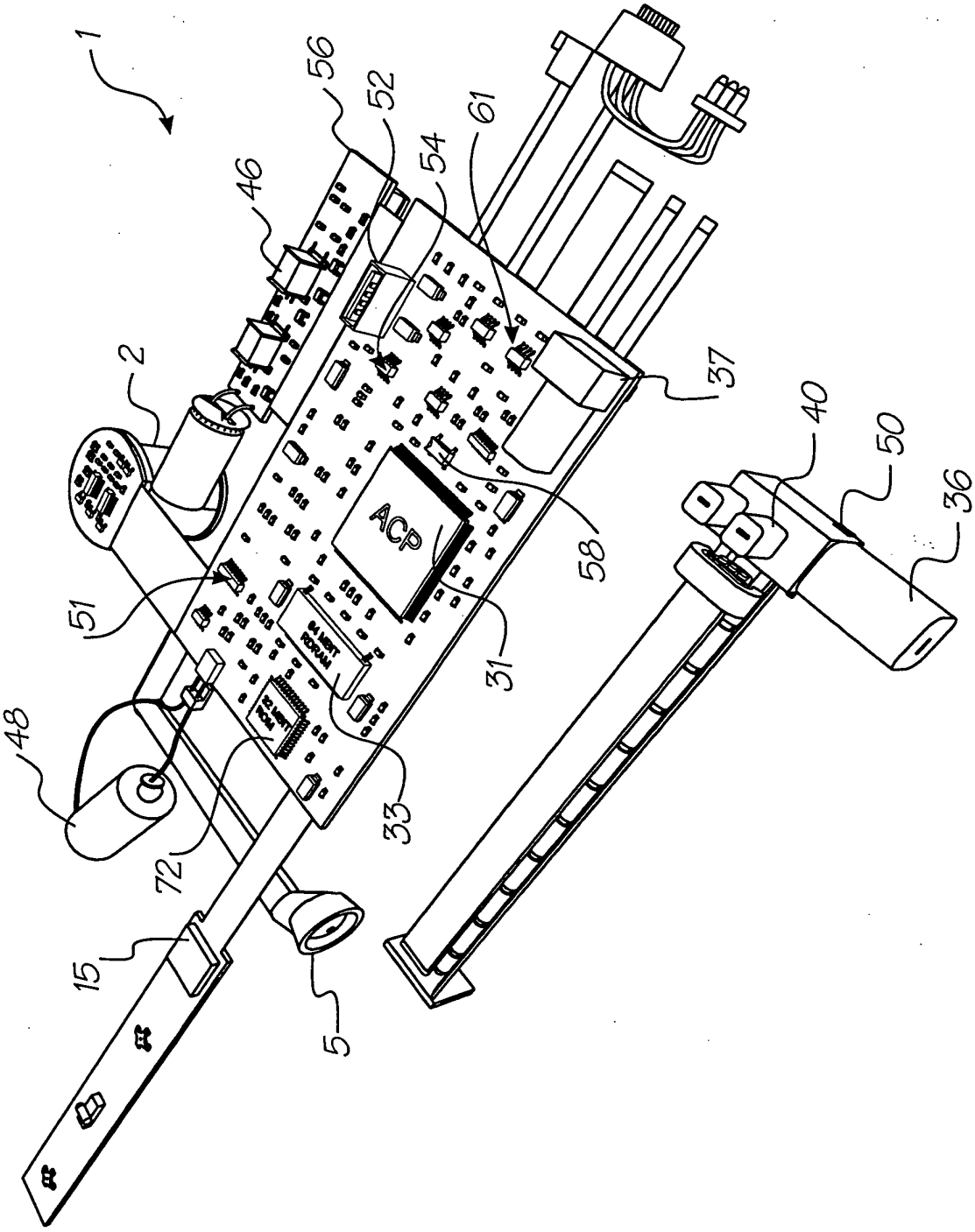


FIG. 213

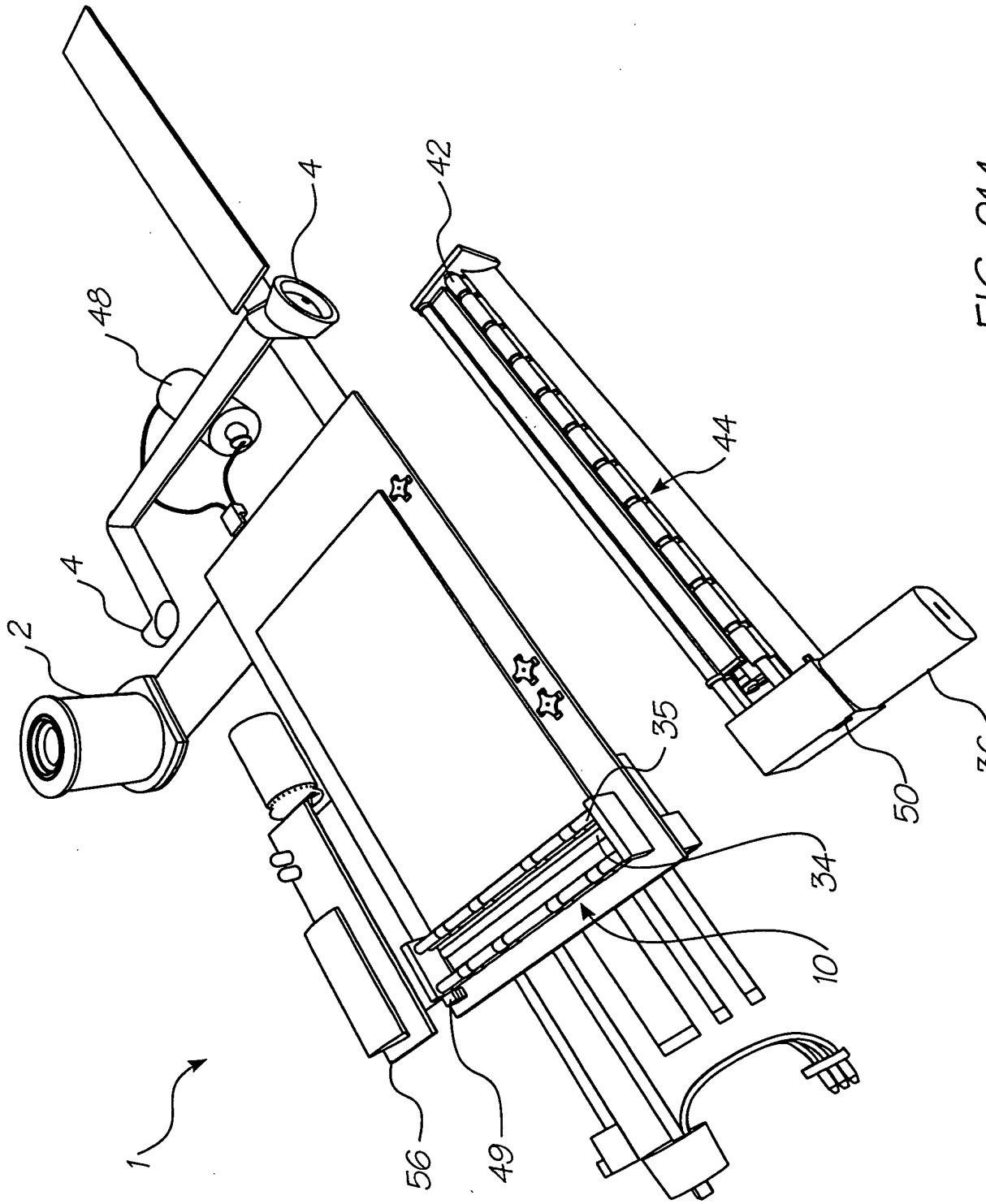


FIG. 214

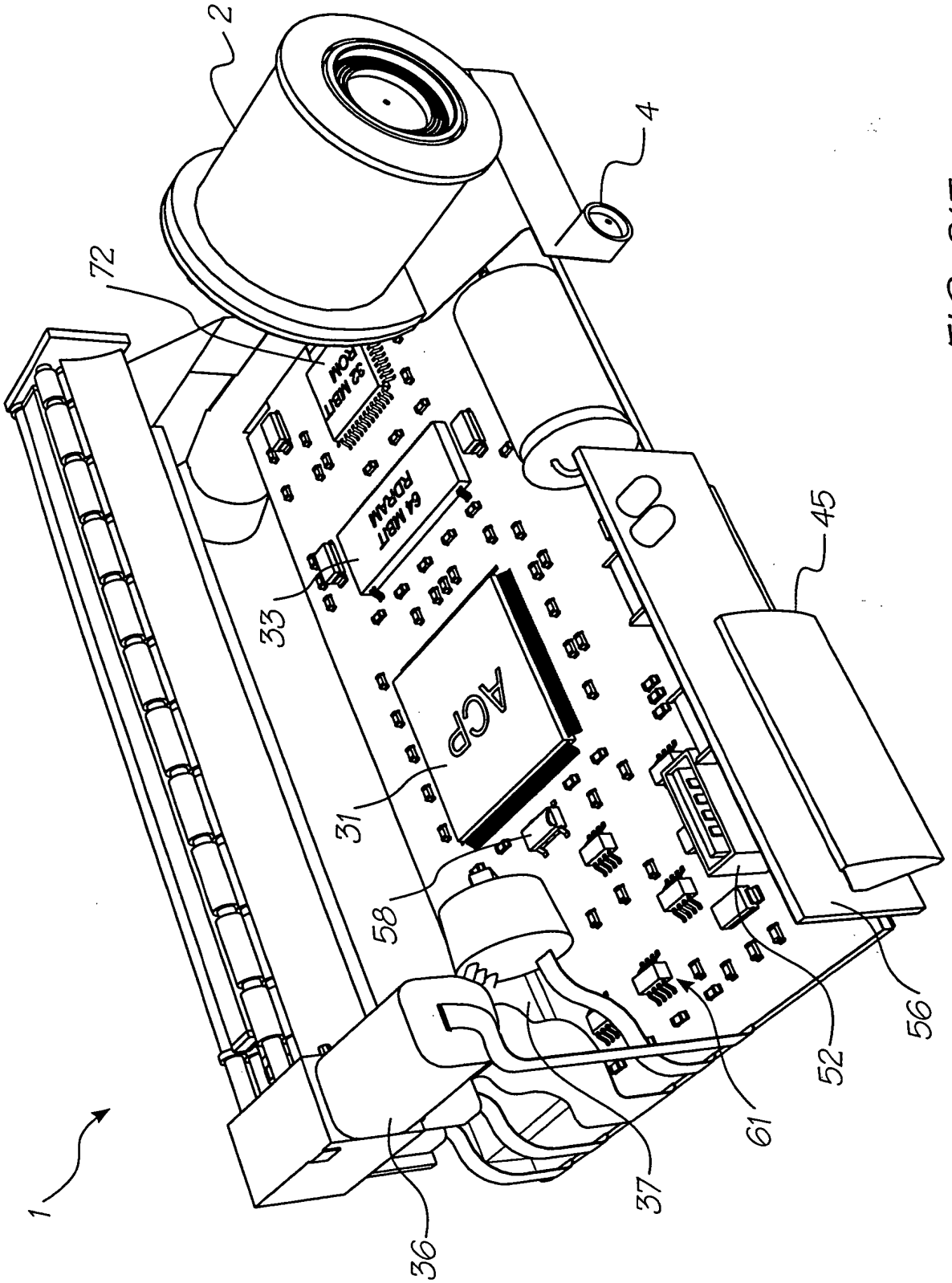


FIG. 215

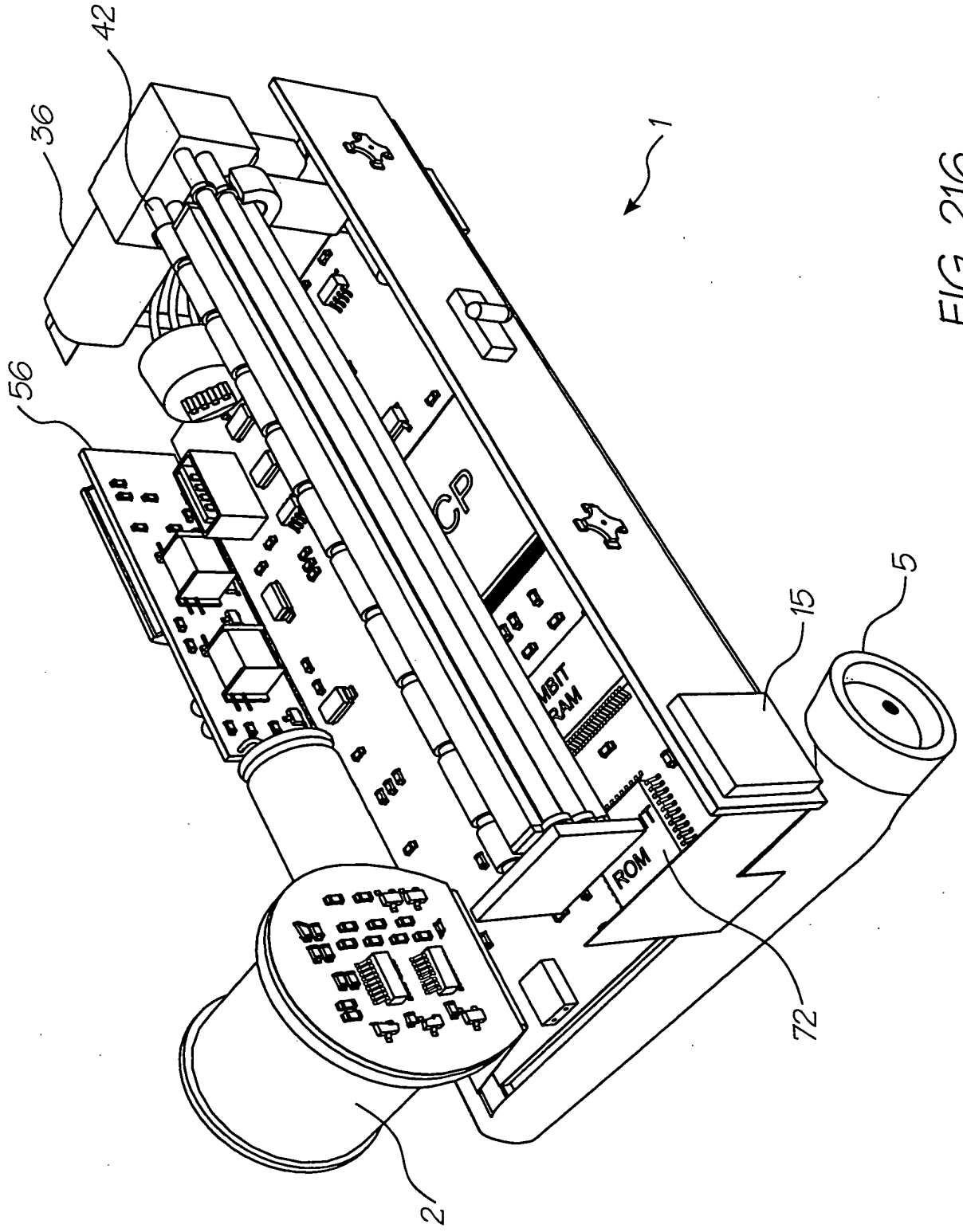


FIG. 216

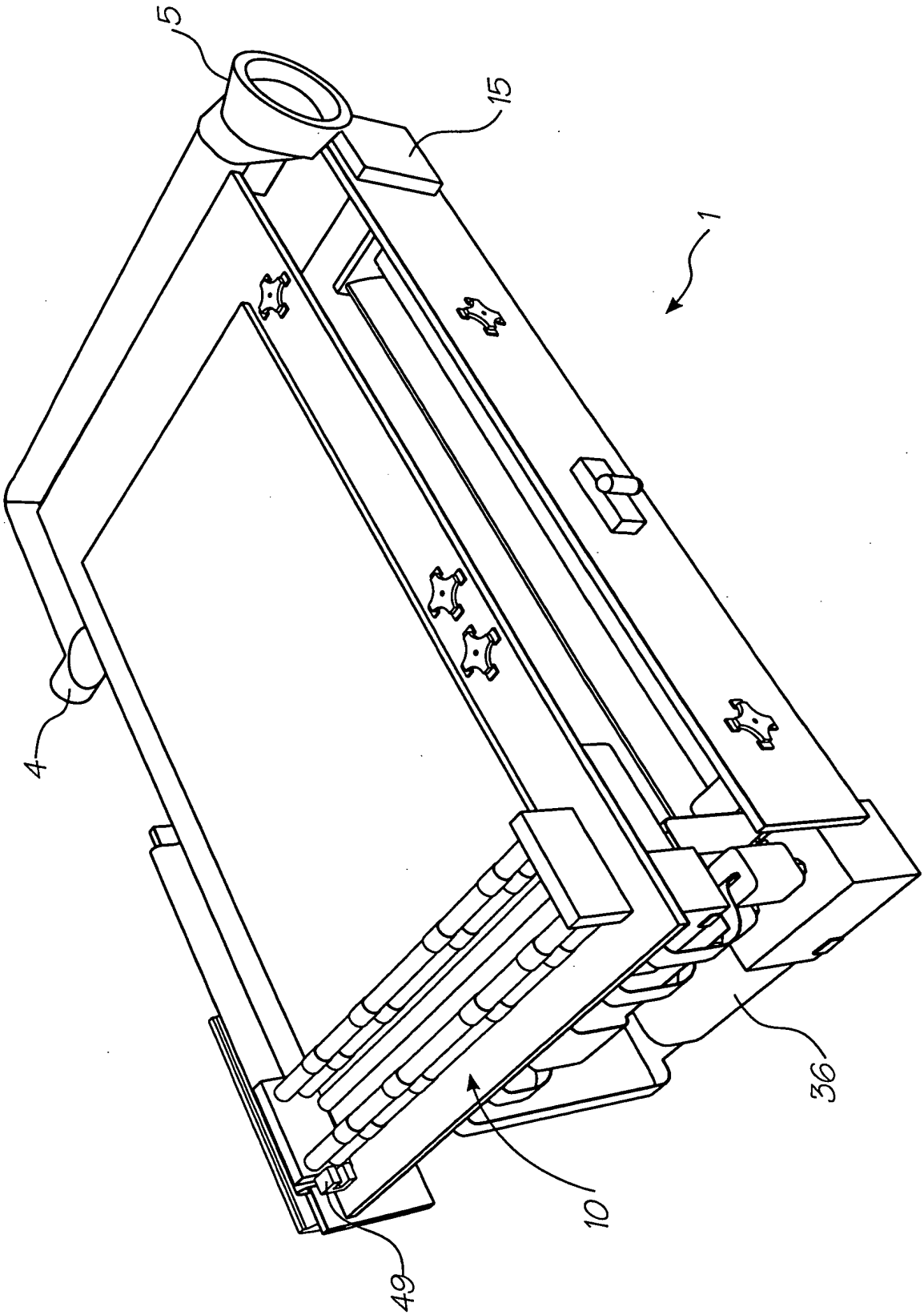


FIG. 217

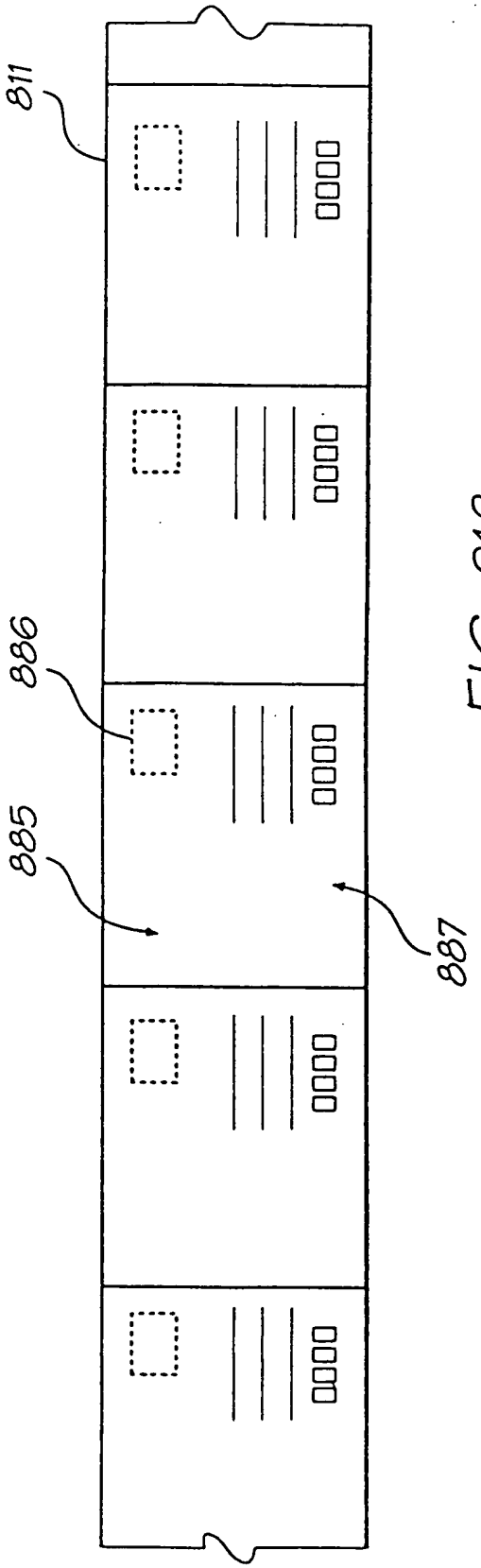


FIG. 218

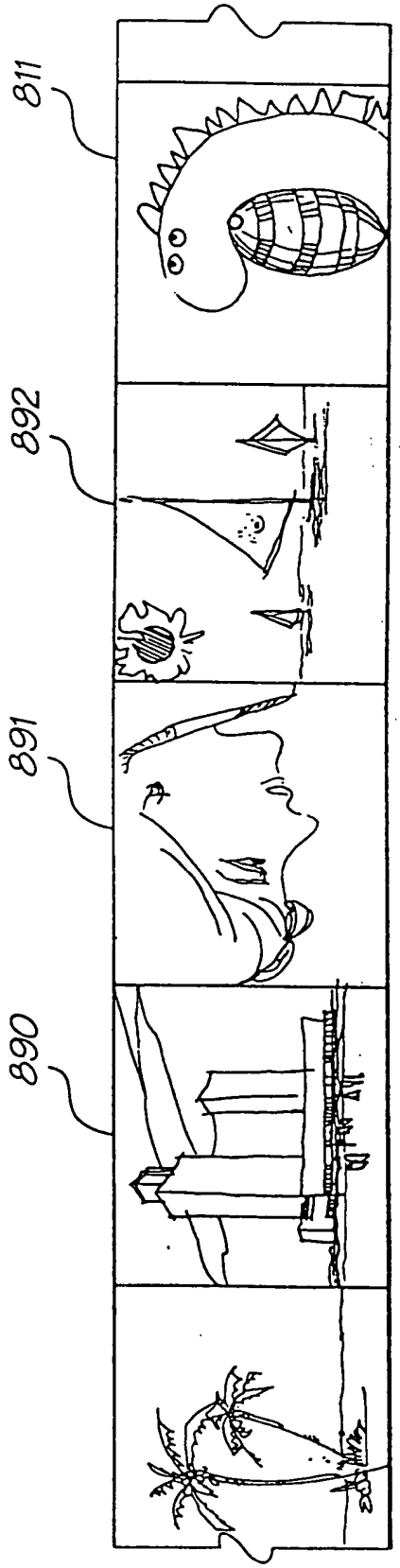


FIG. 219

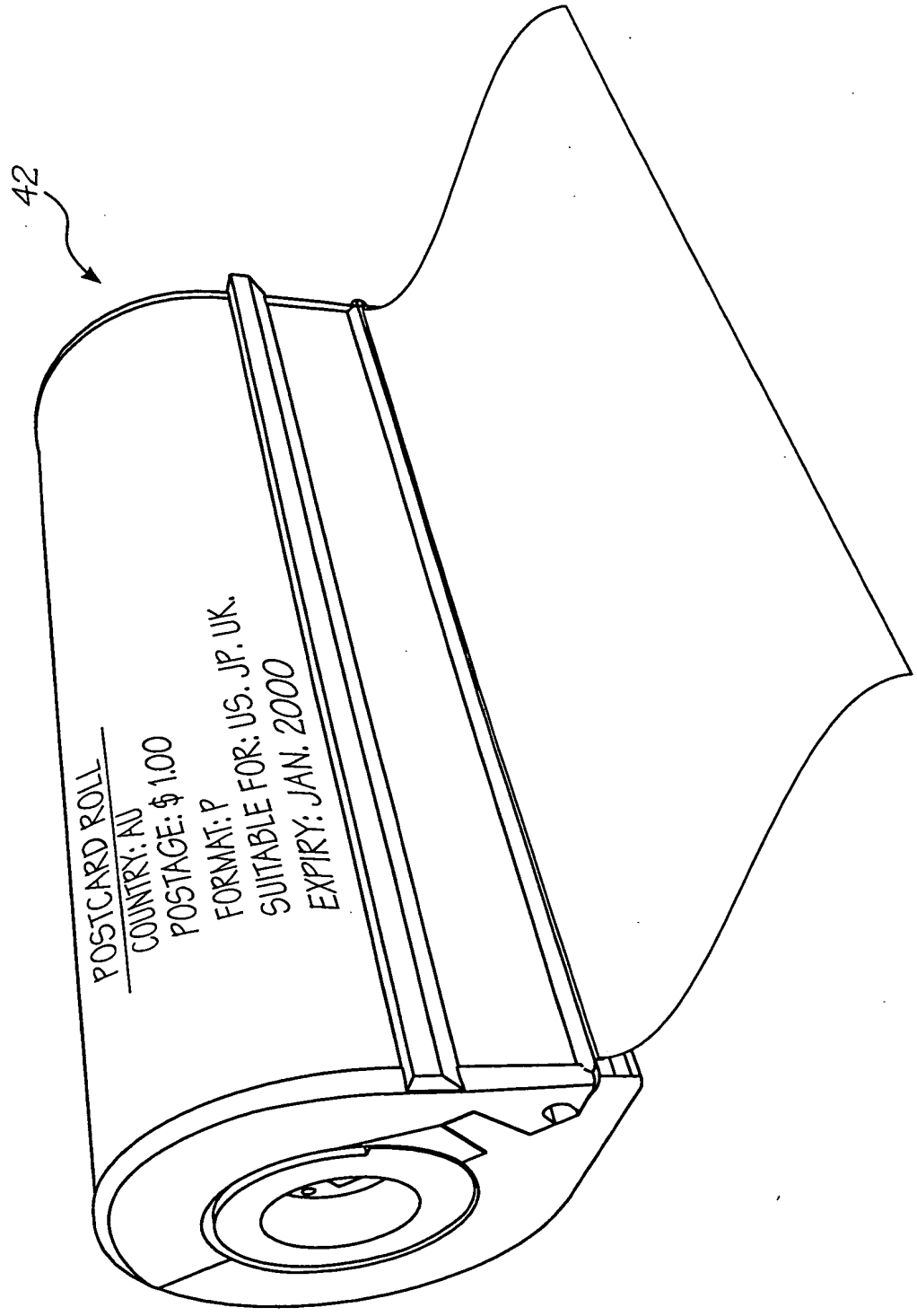

42
POSTCARD ROLL
COUNTRY: AU
POSTAGE: $1.00
FORMAT: P
SUITABLE FOR: US, JP, UK.
EXPIRY: JAN. 2000

FIG. 220

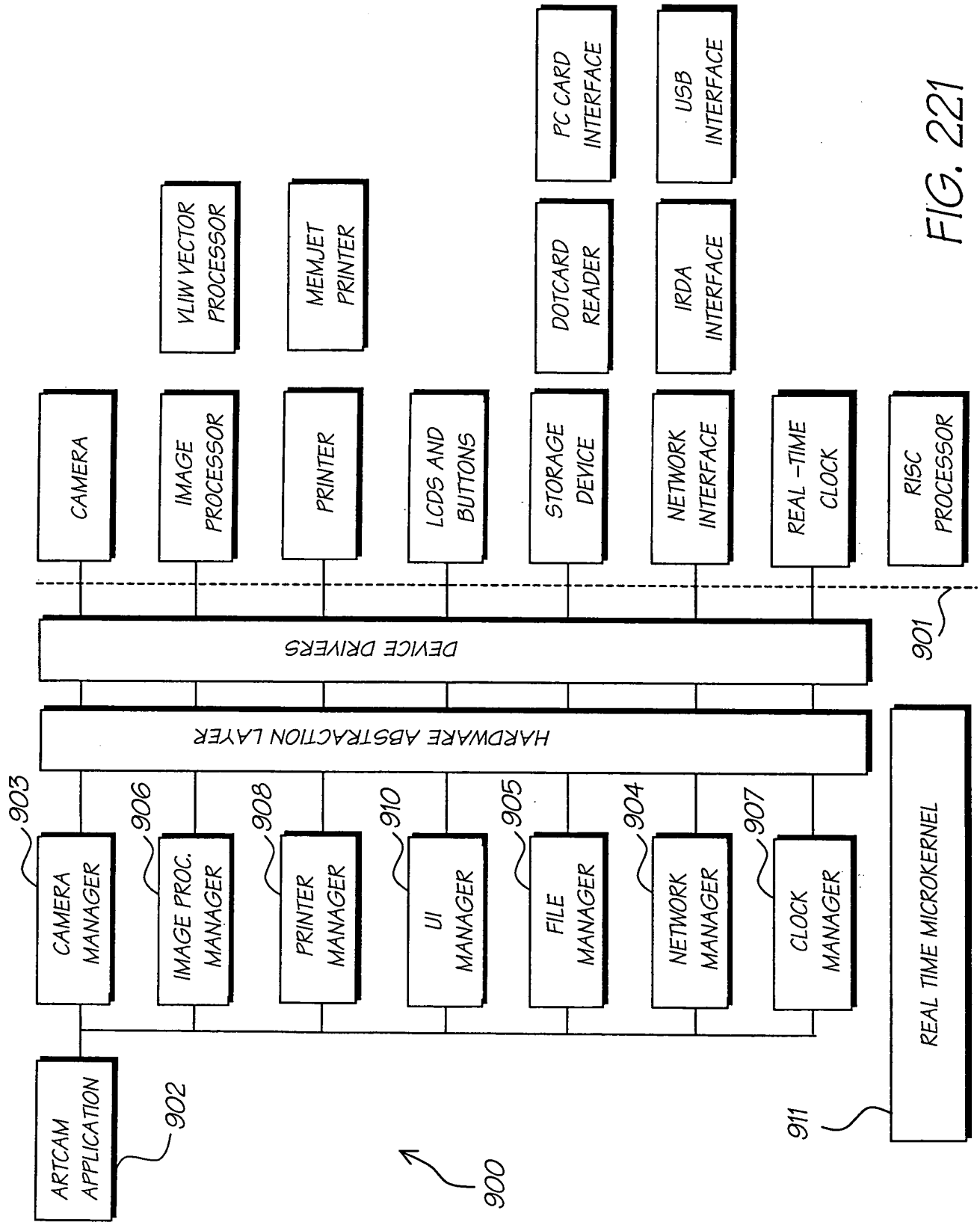

PC CARD INTERFACE
USB INTERFACE
VLIW VECTOR PROCESSOR
MEMJET PRINTER
DOTCARD READER
IRDA INTERFACE
CAMERA
IMAGE PROCESSOR
PRINTER
LCDS AND BUTTONS
STORAGE DEVICE
NETWORK INTERFACE
REAL -TIME CLOCK
RISC PROCESSOR
901
DEVICE DRIVERS
HARDWARE ABSTRACTION LAYER
903
906
908
910
905
904
907
CAMERA MANAGER
IMAGE PROC. MANAGER
PRINTER MANAGER
UI MANAGER
FILE MANAGER
NETWORK MANAGER
CLOCK MANAGER
REAL TIME MICROKERNEL
ARTCAM APPLICATION
902
911
900

FIG. 221

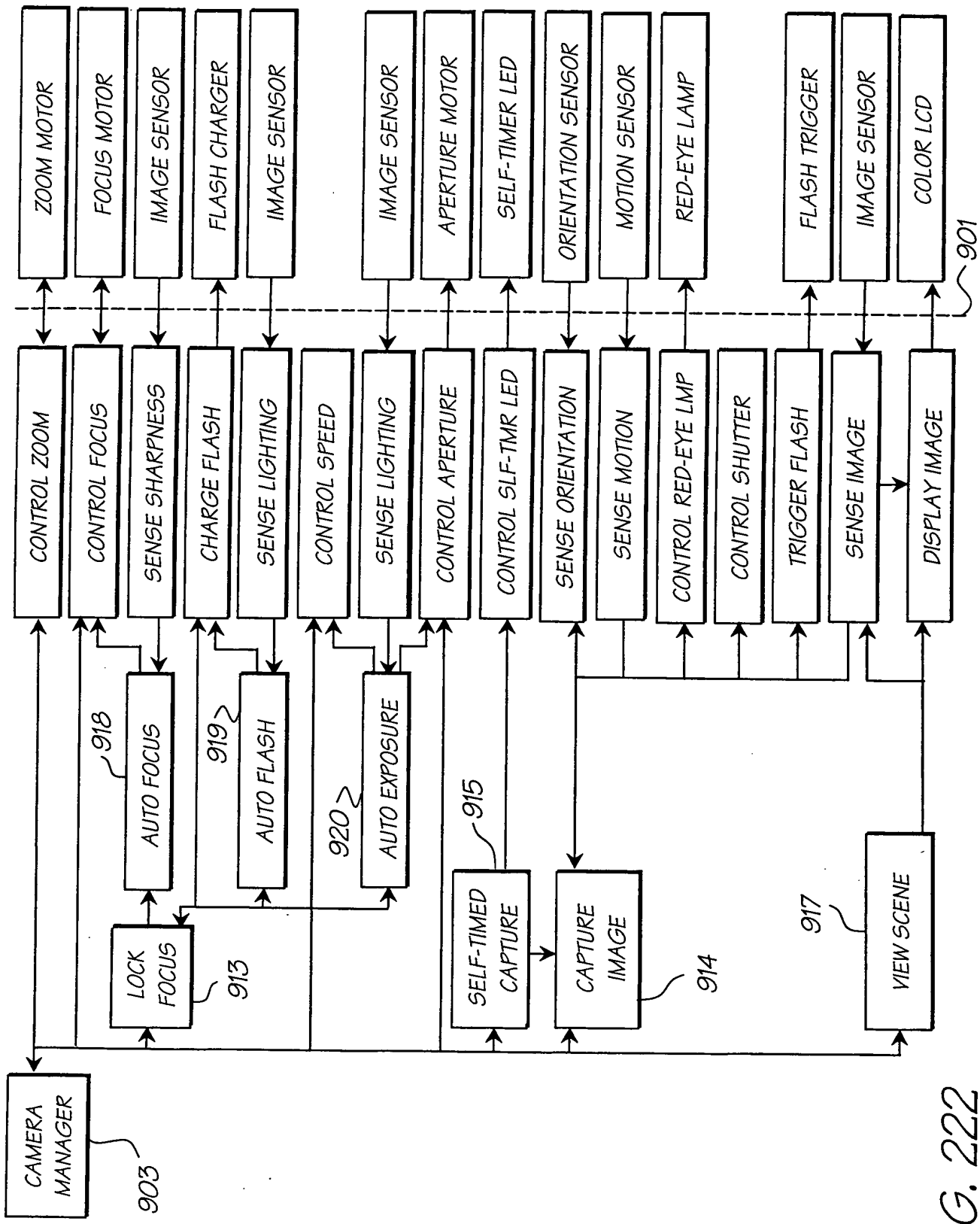

ZOOM MOTOR
FOCUS MOTOR
IMAGE SENSOR
FLASH CHARGER
IMAGE SENSOR
IMAGE SENSOR
APERTURE MOTOR
SELF-TIMER LED
ORIENTATION SENSOR
MOTION SENSOR
RED-EYE LAMP
FLASH TRIGGER
IMAGE SENSOR
COLOR LCD
901
CONTROL ZOOM
CONTROL FOCUS
SENSE SHARPNESS
CHARGE FLASH
SENSE LIGHTING
CONTROL SPEED
SENSE LIGHTING
CONTROL APERTURE
CONTROL SLF-TMR LED
SENSE ORIENTATION
SENSE MOTION
CONTROL RED-EYE LMP
CONTROL SHUTTER
TRIGGER FLASH
SENSE IMAGE
DISPLAY IMAGE
AUTO FOCUS
918
AUTO FLASH
919
AUTO EXPOSURE
920
915
LOCK FOCUS
913
SELF-TIMED CAPTURE
CAPTURE IMAGE
914
917
VIEW SCENE
CAMERA MANAGER
903

FIG. 222

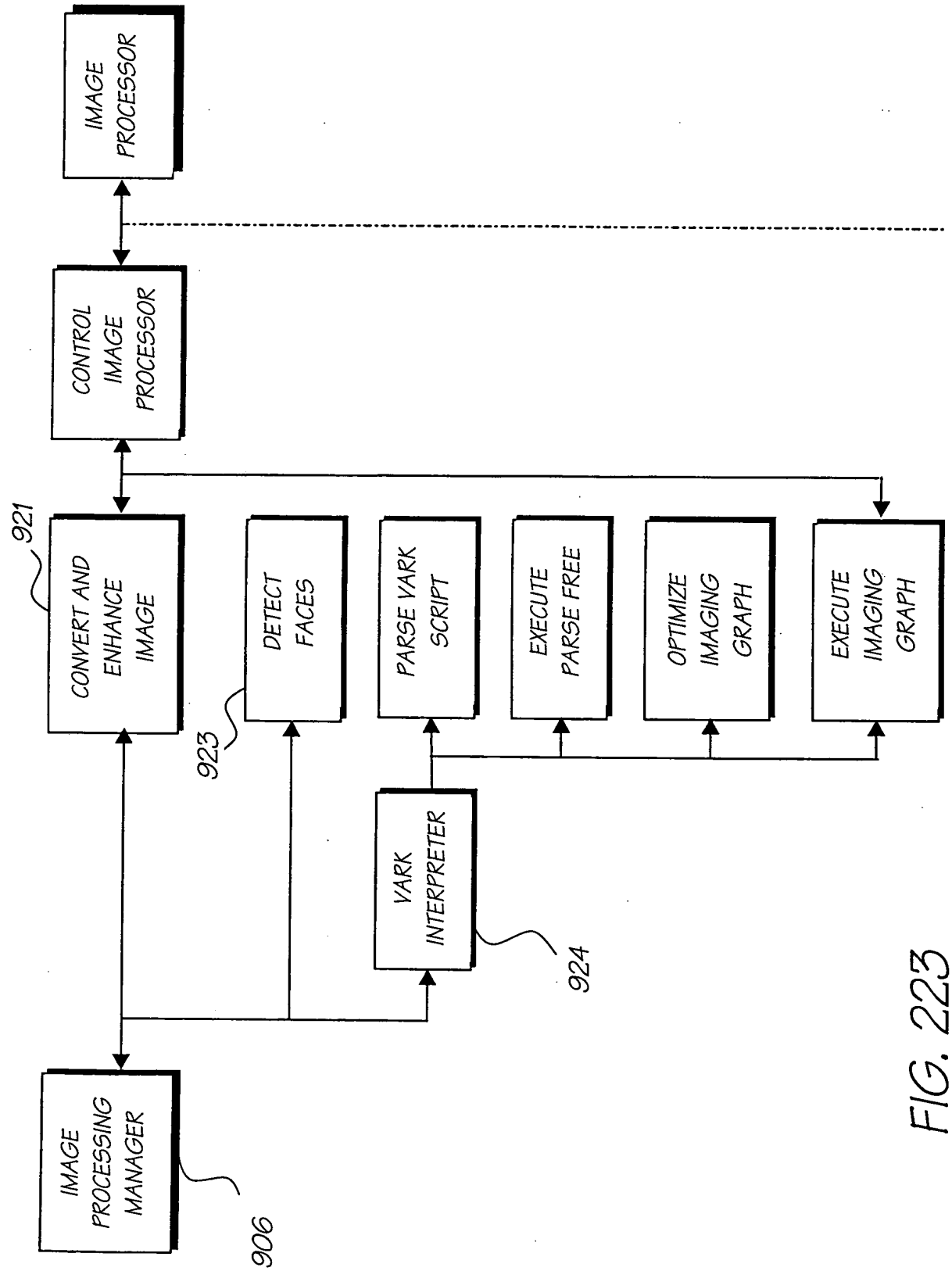


IMAGE PROCESSOR
CONTROL IMAGE PROCESSOR
921
CONVERT AND ENHANCE IMAGE
DETECT FACES
923
PARSE VARK SCRIPT
EXECUTE PARSE FREE
OPTIMIZE IMAGING GRAPH
EXECUTE IMAGING GRAPH
VARK INTERPRETER
924
IMAGE PROCESSING MANAGER
906

FIG. 223

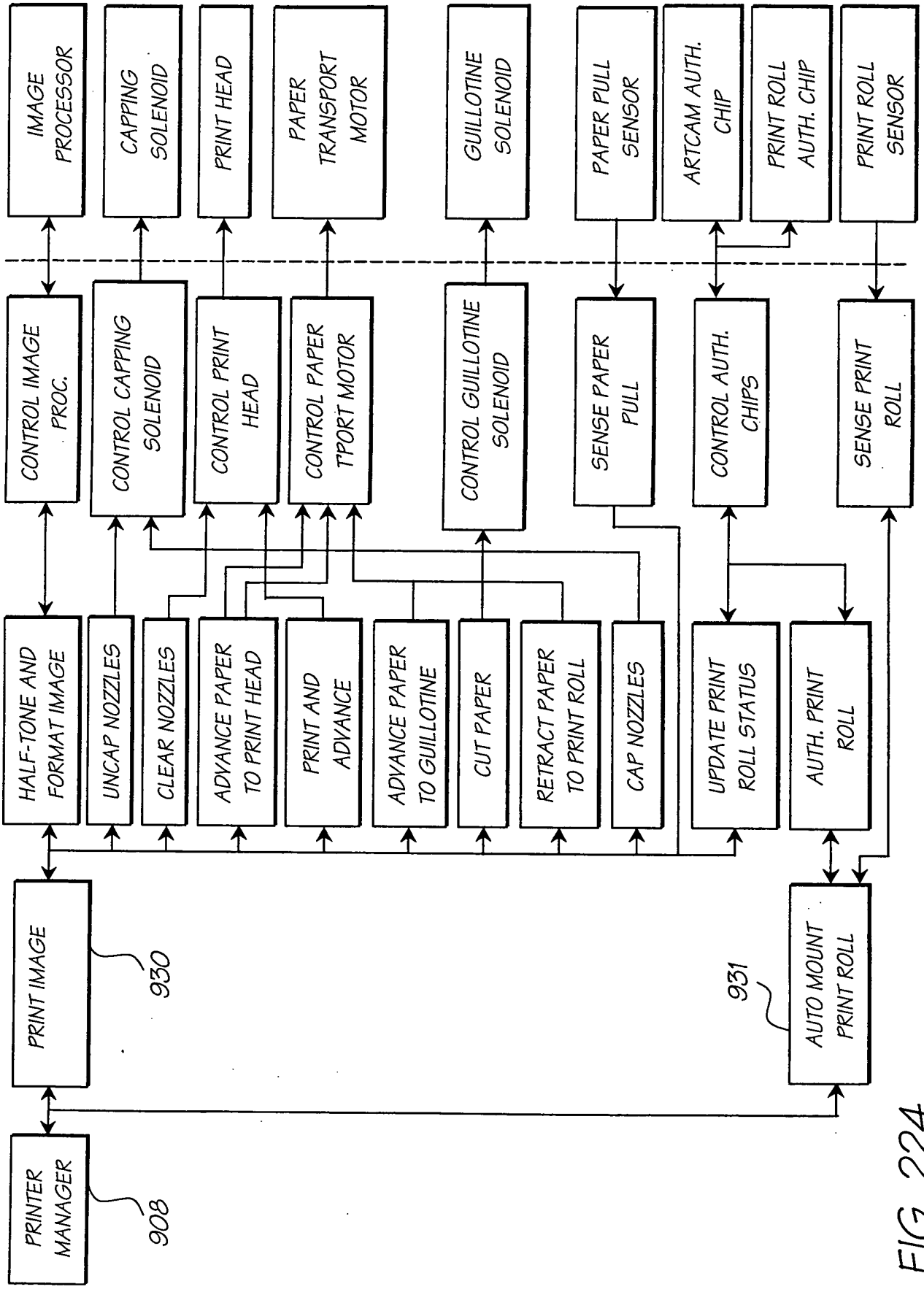


IMAGE PROCESSOR
CAPPING SOLENOID
PRINT HEAD
PAPER TRANSPORT MOTOR
GUILLOTINE SOLENOID
PAPER PULL SENSOR
ARTCAM AUTH. CHIP
PRINT ROLL AUTH. CHIP
PRINT ROLL SENSOR
CONTROL IMAGE PROC.
CONTROL CAPPING SOLENOID
CONTROL PRINT HEAD
CONTROL PAPER TPORT MOTOR
CONTROL GUILLOTINE SOLENOID
SENSE PAPER PULL
CONTROL AUTH. CHIPS
SENSE PRINT ROLL
HALF-TONE AND FORMAT IMAGE
UNCAP NOZZLES
CLEAR NOZZLES
ADVANCE PAPER TO PRINT HEAD
PRINT AND ADVANCE
ADVANCE PAPER TO GUILLOTINE
CUT PAPER
RETRACT PAPER TO PRINT ROLL
CAP NOZZLES
UPDATE PRINT ROLL STATUS
AUTH. PRINT ROLL
PRINT IMAGE
930
AUTO MOUNT PRINT ROLL
931
PRINTER MANAGER
908

FIG. 224

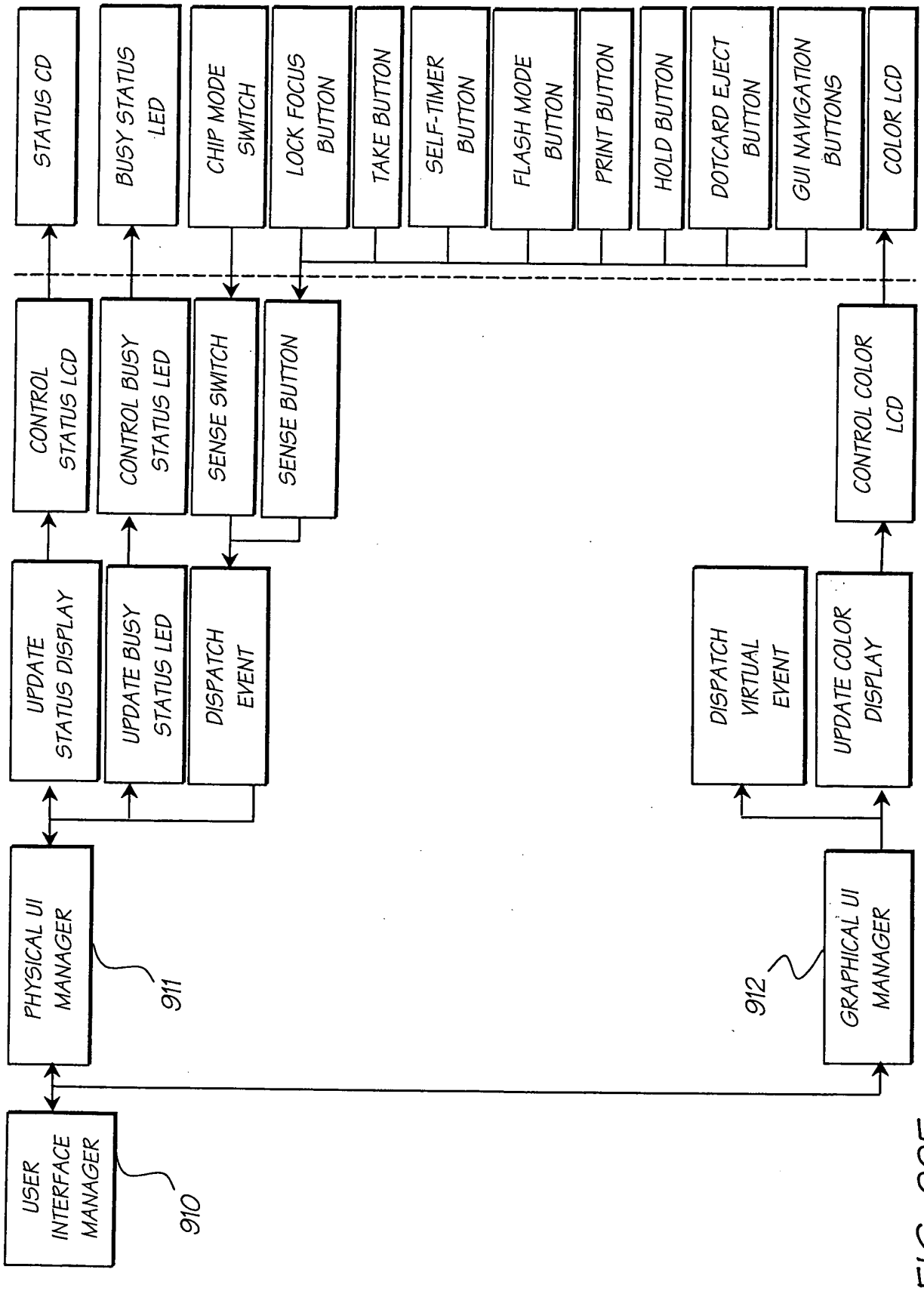

USER INTERFACE MANAGER
910
PHYSICAL UI MANAGER
911
GRAPHICAL UI MANAGER
912
UPDATE STATUS DISPLAY
UPDATE BUSY STATUS LED
DISPATCH EVENT
DISPATCH VIRTUAL EVENT
UPDATE COLOR DISPLAY
CONTROL STATUS LCD
CONTROL BUSY STATUS LED
SENSE SWITCH
SENSE BUTTON
CONTROL COLOR LCD
STATUS CD
BUSY STATUS LED
CHIP MODE SWITCH
LOCK FOCUS BUTTON
TAKE BUTTON
SELF-TIMER BUTTON
FLASH MODE BUTTON
PRINT BUTTON
HOLD BUTTON
DOTCARD EJECT BUTTON
GUI NAVIGATION BUTTONS
COLOR LCD

FIG. 225

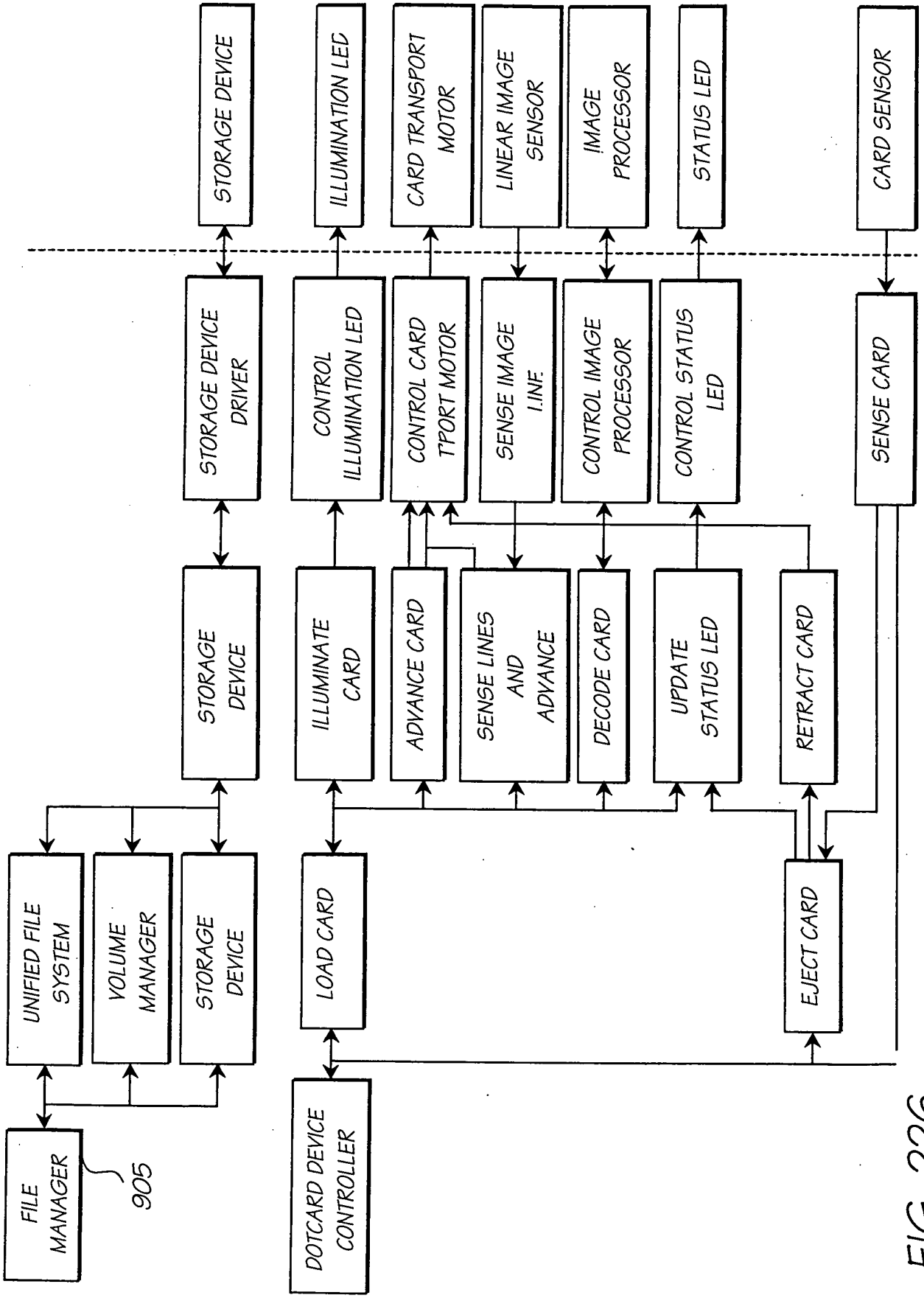

STORAGE DEVICE
ILLUMINATION LED
CARD TRANSPORT MOTOR
LINEAR IMAGE SENSOR
IMAGE PROCESSOR
STATUS LED
CARD SENSOR
STORAGE DEVICE DRIVER
CONTROL ILLUMINATION LED
CONTROL CARD TPORT MOTOR
SENSE IMAGE LINE
CONTROL IMAGE PROCESSOR
CONTROL STATUS LED
SENSE CARD
STORAGE DEVICE
ILLUMINATE CARD
ADVANCE CARD
SENSE LINES AND ADVANCE
DECODE CARD
UPDATE STATUS LED
RETRACT CARD
UNIFIED FILE SYSTEM
VOLUME MANAGER
STORAGE DEVICE
LOAD CARD
EJECT CARD
FILE MANAGER
905
DOTCARD DEVICE CONTROLLER

FIG. 226

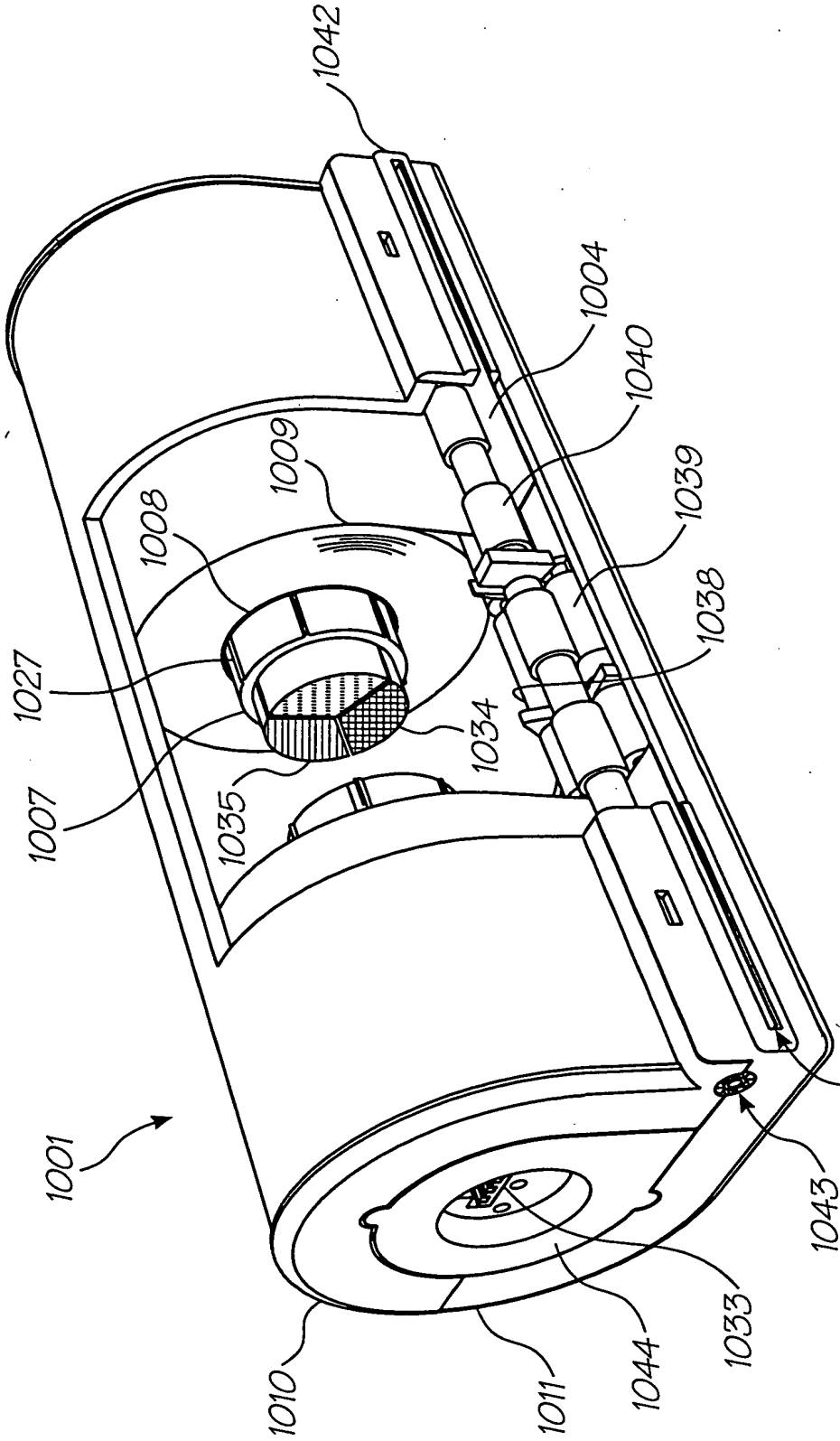

1042
1004
1040
1039
1038
1009
1008
1027
1007
1035
1034
1001
1010
1011
1044
1033
1043
1012

FIG. 227

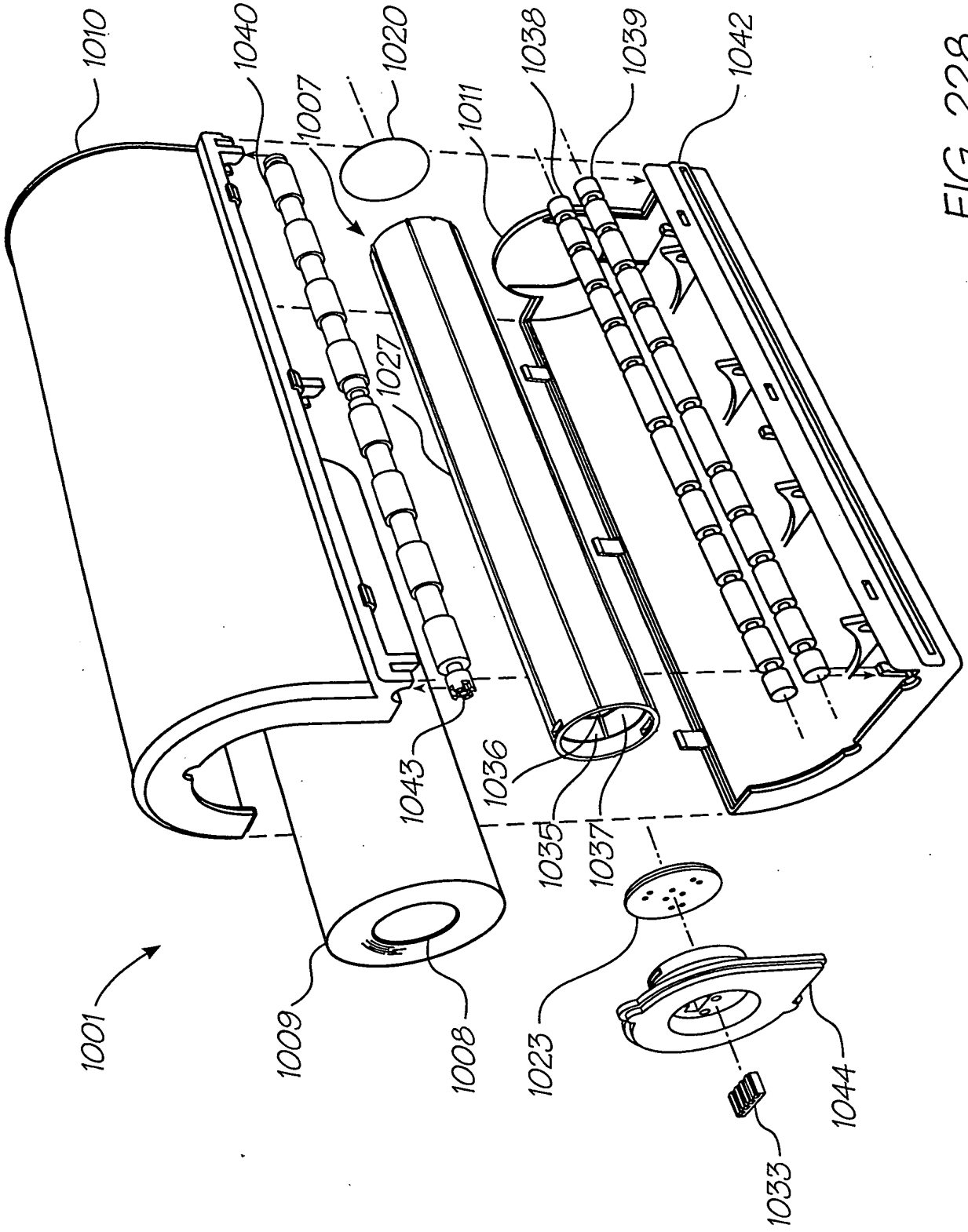

1010
1040
1007
1020
1011
1038
1039
1042
1027
1043
1036
1035
1037
1001
1009
1008
1023
1044
1033

FIG. 228

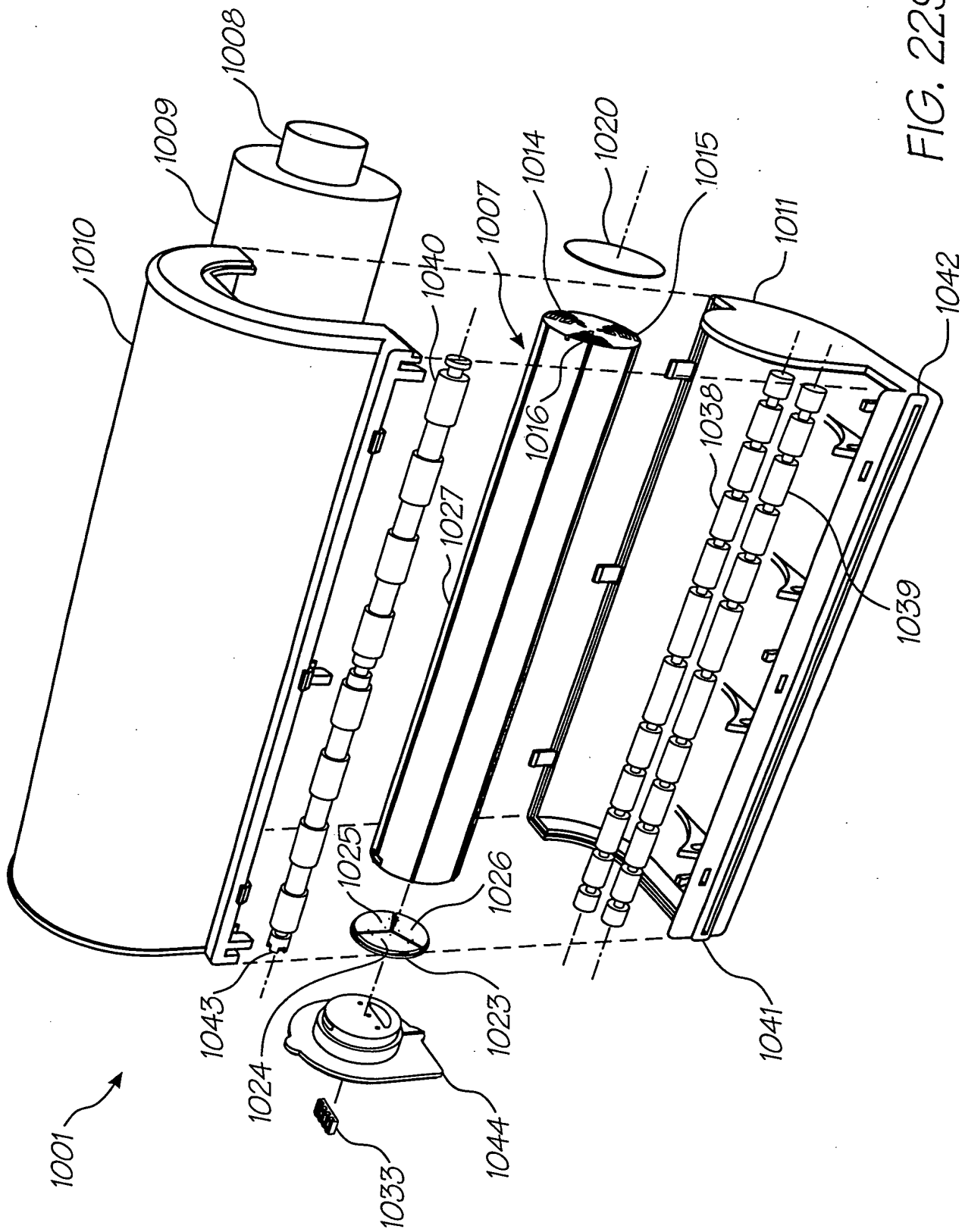

1001
1008
1009
1010
1040
1007
1014
1020
1015
1011
1042
1016
1038
1039
1027
1025
1026
1043
1024
1023
1041
1044
1033

FIG. 229

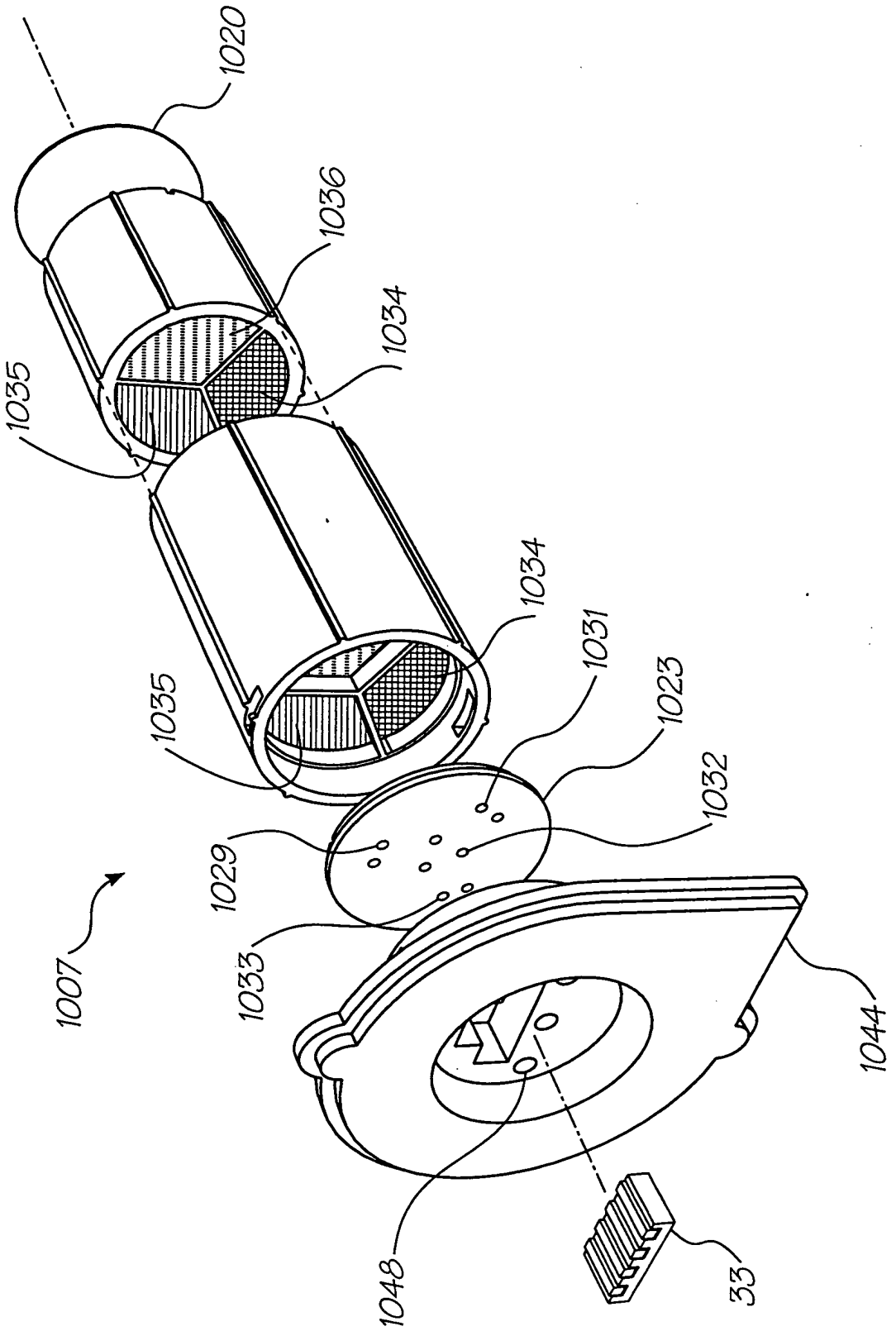

1020
1036
1034
1035
1034
1031
1023
1035
1032
1029
1007
1033
1044
1048
33

FIG. 230

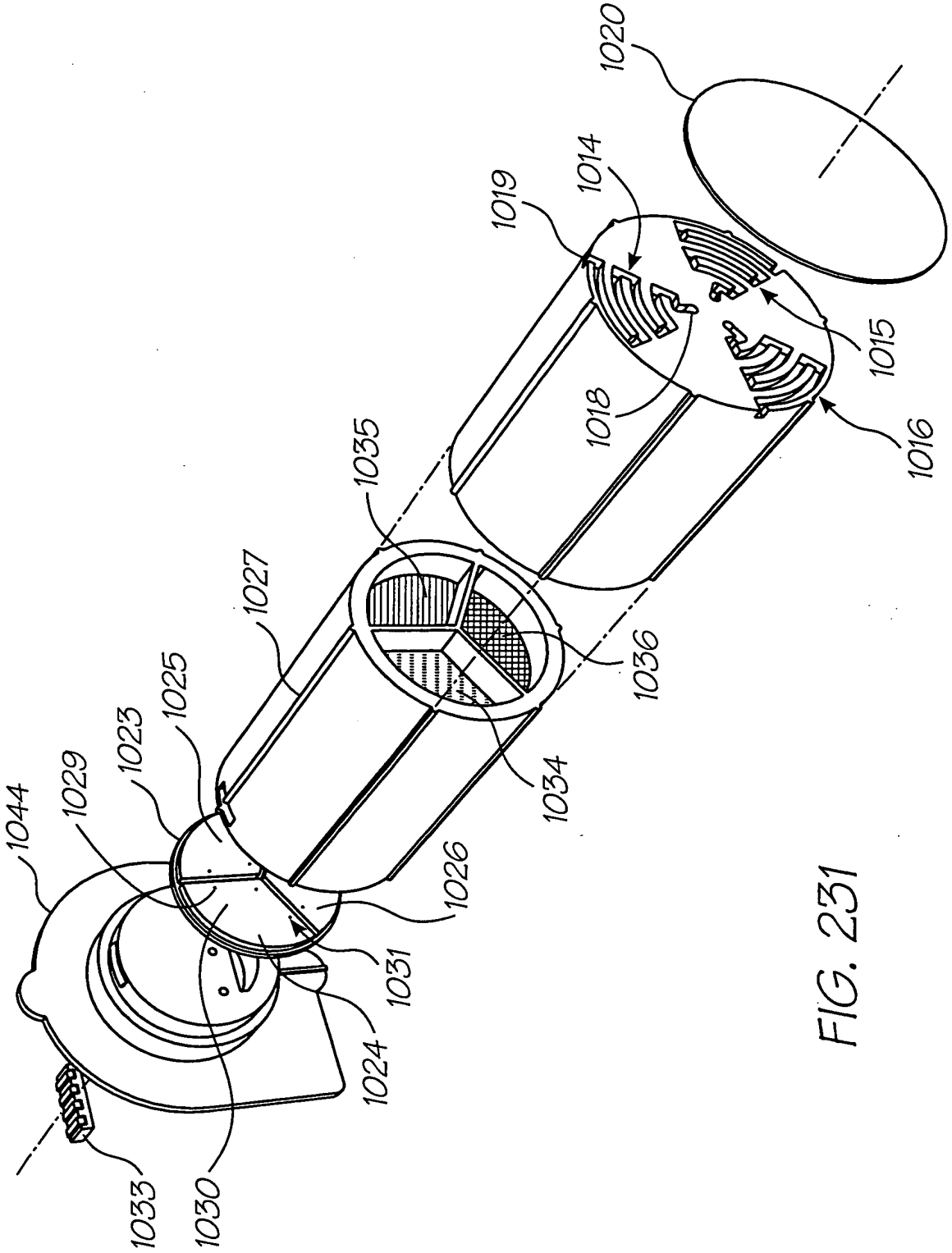

1020
1014
1019
1015
1018
1016
1035
1027
1036
1025
1023
1034
1029
1044
1026
1031
1024
1033
1030

FIG. 231